SANDIA REPORT
SAND2008-6094
Unlimited Release
September, 2008

Foam Process Models

Rekha R, Rao, Lisa A. Mondy, Harry K. Moffat, David R. Noble, Patrick K. Notz, Douglas B. Adolf, Thomas A. Baer

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185-0836

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2008-6094
Unlimited Release
Printed September 2008

Foam Process Models

Rekha R. Rao, Lisa A. Mondy, David R. Noble
Thermal Fluid Processes

Harry K. Moffat
Nanoscale and Reactive Processes

Patrick K. Notz
Computational Thermal and Fluid Mechanics

Douglas B Adolf
Organic Materials
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS0836

Thomas A. Baer
Modeling, Simulation and Analysis Department
Procter & Gamble Co.
Beckett Ridge – CETL- IP 332
8256 Union Centre Blvd.
West Chester, OH 45069

Abstract

In this report, we summarize our work on developing a production level foam processing computational model suitable for predicting the self-expansion of foam in complex geometries. The model is based on a finite element representation of the equations of motion, with the movement of the free surface represented using the level set method, and has been implemented in SIERRA/ARIA. An empirically based time- and temperature-dependent density model is used to encapsulate the complex physics of foam nucleation and growth in a numerically tractable model. The change in density with time is at the heart of the foam self-expansion as it creates the motion of the foam. This continuum-level model uses an homogenized description of foam, which does not include the gas explicitly. Results from the model are compared to temperature-instrumented flow visualization experiments giving the location of the foam front as a function of time for our EFAR model system.

ACKNOWLEDGMENTS

We appreciate the funding provided by ASC to develop this model, implement it in SIERRA/ARIA, and write this SAND report. The model would be nothing without the experimental team who made experimental discoveries for model development, determined parameters to populate the model, and provided validation data. The experimental team, in addition to authors Lisa Mondy and Doug Adolf, includes Anne Grillet, Ed Russick, Chris Brotherton, Chris Bourdon, Allen Gorby, Ray Cote, Jaime Castañeda and Kyle Thompson. Thanks especially to Anne and Ed for jumping in and getting results at the 11th hour. Andy Kraynik has been invaluable for pointing us in the right direction for correlations from the foam literature and his general knowledge of foam. Jim Mahoney from Kansas City Plant is our muse and has inspired this project by politely asking for a foam self-expansion modeling tool since none are commercially available at the present time. He has also tirelessly provided meshes and geometries, adding and changing the boundary conditions at my insistence.

CONTENTS

FIGURES

TABLES

NOMENCLATURE

ASC	Advanced Strategic Computing
AF&F	Arming, Fuzing, and Firing
C6	Campaign 6
DOE	Department of Energy
DSC	Differential Scanning Calorimeter
EFAR	Epoxy Foam Able Replacement
KCP	Kansas City Plant
REF	Removable Epoxy Foam
SNL	Sandia National Laboratories

1. INTRODUCTION

1.1. Background

Foams are ubiquitous low density materials used for a variety of applications including shock, thermal, and vibration isolation of electronic components, disposable containers, and energy production. Foams are used widely in the weapons' complex as encapsulants to protect sensitive electronic components from shock and vibration. Despite their many uses, foams are still not well understood at a fundamental level [Seo et al., 2003; Mao et al, 2006]. Two major categories of foam exist: chemically blown foams and physically blown foams. Chemically blown foams expand via reactions that produce a gas phase during polymerization, e.g. polyurethanes, while physical blown foams begin with a dissolved blowing agent that boils to produce cells either by increasing the temperature or decreasing the pressure. For our applications, we are interested in blown foams, such as EFAR (Epoxy Foam Able Replacement) and REF (Removable Epoxy Foam) used to encapsulate portions of the AF&F. These foams start off as an emulsion of fluorinert blowing agent in epoxy monomer and curative. Once this emulsion is formed, the foam precursor is injected into the mold and inserted into an oven to boil the fluorinert and produce foam. The complex interplay between heat transfer, polymerization, boundary conditions and nucleation of fluorinert can make predetermination of the final foam density and amount needed to fill the mold difficult. The goal of this work is to use the data provided by the Campaign 6 (C6) Foam Processing Project to better our understanding of the physical process of epoxy foam blowing and to use that understanding to build an engineering model, which, in turn, can be used to predict the foaming process. This model, in turn, will be used to address processing issues in foam such as voids, incomplete embedding, and inhomogeneous properties and microstructure that can lead to production delays and potential SFIs. Beyond troubleshooting, the model can also improve foam encapsulation by optimizing gate and vent locations, material properties, and processing conditions.

In this report, we summarize our work on developing a production level, foam processing computational model suitable for predicting the self-expansion of foam in complex geometries. The model is based on a finite element representation of the equations of motion, with the movement of the free surface represented using the level set method and has been implemented into the SIERRA/ARIA mechanics code. An empirically based time- and temperature-dependent density model is used to encapsulate the complex physics of foam nucleation and growth in a numerically tractable model. The change in density with time is at the heart of the foam self-expansion as it creates the motion of the foam. This continuum-level model uses an homogenized description of foam, which does not include the gas explicitly, but only through the density model. The viscosities, shear and bulk, are a complex function of temperature, degree of polymerization, and gas bubble volume fraction. For our epoxy foams, foam processing is non-isothermal because of the low temperature mixing step required by the physical blowing agent and the heat increase associated with the exothermic polymerization. Here, we use thermal properties that vary with gas volume fraction as well. When available, physical properties are determined from experimental data obtained from the C6 Foam Processing Project.

In addition to varying local fields such as temperature and gas volume fraction, the material models also vary with the location of the level set interface taking thermal and fluid properties of the displaced air phase in the negative level set region and those of the foam in the positive level set region. The level set zero describes the location of the interface, where surface forces are applied using the Continuous Surface Force (CSF) treatment. The variation from foam to gas properties is handled with a diffuse interface method using a numerical Heaviside function. Results from the model are compared to temperature-instrumented flow visualization experiments giving the location of the foam front as a function of time for our EFAR model system, used because it is more readily available that the REF.

Though our work herein focuses on EFAR, it should also be applicable to REF, and other foam encapsulants including chemically blown foams such as PMDIs, once the models are populated with parameter values suitable to those materials. Part of the C6 project was focused on creating a methodology to characterize foams that can be applied to other systems of interest.

1.2. Composition of EFAR and REF, Mixing and Foaming Protocols

The EFAR20 foam is initially mixed as a Part A and Part B mixture, where 20 indicates the recipe for a 20lb/ft^3 foam. (The recipe for EFAR08 is similar, but contains more fluorinert to produce an 8lb/ft^3 foam.) The detailed weight fractions and density of each component are given in Table 1. For completeness, we also include information for REF08 though this foam is not used in this study.

Table 1. Epoxy Foam Formulations Used in this Study

Material	**Weight fraction**	**Density (g/cm^3)**
EFAR20		**(Unfoamed) 1.14**
EFAR20 Part A (resin)	*0.654*	*1.17*
• Shell Epon 828 resin	• 0.6	• 1.17
• Shell Epon 8121 resin	• 0.4	• 1.17
EFAR20 Part B (curative)	*0.346*	*1.08*
• Air Products Ancamine 2049 curing agent	• 0.585	• 0.95
• Shell Epi-Cure 3270 curing agent	• 0.245	• 0.97
• Air Products DC-193 surfactant	• 0.019	• 1.07
• Cabot Cab-O-Sil M-5 fumed silica	• 0.019	• 2.20
• 3M Fluorinert FC-72	• 0.132	• 1.7
REF308		**(Unfoamed) 1.23**
REF308 Part A (resin)	*0.627*	*1.19*
• Removable epoxy resin 1 (RER1)	• 0.48	• 1.2, approx.
• Removable resin 2 (RR2)	• 0.12	• 1.2, approx.
• Shell Epon 8121 resin	• 0.40	• 1.17
REF308 Part B (curative)	*0.373*	*1.23*
• Air Products Ancamine 2049 curing agent	• 0.361	• 0.95
• Air Products Ancamine 2205 curing agent	• 0.142	• 1.04
• Air Products DC-193 surfactant	• 0.069	• 1.07

• Cabot Cab-O-Sil M-5 fumed silica • 3M Fluorinert FC-72	• 0.009 • 0.419	• 2.20 • 1.7

For both EFAR and REF, Part A is preheated to the standard oven temperature of 65°C, while Part B remains at room temperature to insure that the fluorinert does not boil prematurely. The boiling temperature of fluorinert in Albuquerque, at altitude, is 53°C and at Kansas City, at sea level, is 56°C; thus the fluorinert is superheated relative to its boiling temperature in a 65°C oven.

Part A and B are mixed vigorously to form an emulsion of fluorinert droplets in an epoxy continuous phase. An additional effect of the mixing is that a significant amount of air is entrained into the resulting mixture. The air phase plays an important role in nucleating the boiling of the liquid fluorinert [Mondy et al, 2008]. Figure 1 shows a micrograph of the foam precursor, with the fluorinert droplets having a size of about 10μm while the air bubbles have a size of roughly 100μm.

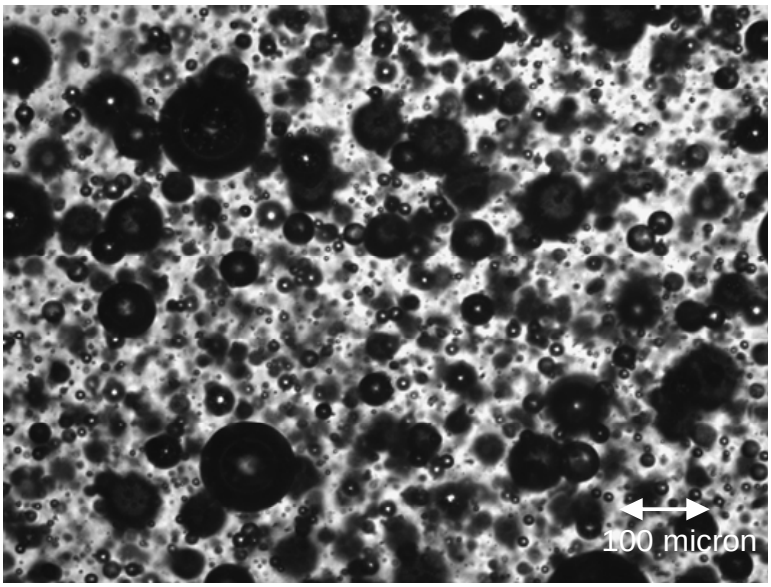

Figure 1. Epoxy foam starts out as an emulsion and probably nucleates heterogeneously.

The foam precursor takes on a temperature of 43°C after mixing the 65°C Part A with the room Part B, kept at a low temperature to keep the fluorinert from boiling. This mixture is then injected into a preheated mold and a preheated oven at 65°C. The material foams to fill the mold in roughly 5 minutes, depending on the mold geometry. After an hour in the mold, the oven temperature is increased to 75°C to increase the rate of polymerization.

2. CONTINUUM MODEL

2.1. Continuum- Level Model for Foaming Materials

Physically blown foams are inherently two-phase materials with bubbles of air and fluorinert being surrounded by a continuous phase of epoxy. A detailed mesoscale model including the dynamics, deformation, mass transfer, and pressurization of each bubble, of which there are millions, would be ideal but is beyond the scope of current computational techniques. Here, instead, we develop a continuum-level model that uses averaged properties between the gas bubbles and the liquid phase to determine the behavior of the foam during the self-expansion process. This self-expansion process is driven by the phase change of the liquid fluorinert to gas due to oven heating. Here we encapsulate the complex physics of nucleation and boiling into a time- and temperature-dependent density function, which decreases in time creating velocity through the continuity equation. The continuity equation is written to emphasize the change in density as the source of foam velocity generation, where v is the mass averaged velocity and ρ is the foam density.

$$\nabla \bullet v = -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial t} + v \bullet \nabla \rho\right) \tag{1}$$

Conservation of momentum takes into account gradients in the fluid stress, τ, and pressure, p, as well as gravitational effects. Note that gravity is applied to the homogenized foam material and does not take into account the buoyancy differences between the epoxy and the gas bubbles.

$$\rho \frac{\partial v}{\partial t} + \rho v \bullet \nabla v = \nabla \bullet \underline{\underline{\tau}} - \nabla p + \rho \underline{g} \tag{2}$$

The stress tensor has a generalized Newtonian shear viscosity, in addition to a generalized Newtonian bulk viscosity. The bulk viscosity is associated with the fact that the divergence of the velocity field is non-zero and we have a dilatational flow [Bird et al., 1960]. The bulk viscosity term produce only normal stresses and not shear stresses.

$$\begin{aligned} \underline{\underline{\tau}} &= \eta(\nabla v + \nabla v^t) - \left(\frac{2}{3}\eta - \kappa\right)(\nabla \bullet v)\underline{\underline{I}} \\ \underline{\underline{\pi}} &= \underline{\underline{\tau}} - p\underline{\underline{I}} \end{aligned} \tag{3}$$

It is a useful construct to define a total stress, π, based on the fluid stress and the pressure, as we will see in subsequent sections. The generalized Newtonian viscosity models imply that the viscosities vary with local fields, but still have a Newtonian form where stress is proportional to strain. However, the gradients of η and κ are non-zero and must be included in the momentum equation. Here, both the shear and bulk viscosities are a function of temperature, degree of polymerization, and gas bubble volume fraction.

Once the stress tensor is substituted into the momentum equation, we obtain the following equation which contains the usual Newtonian terms plus the bulk viscosity and gradients of the viscosity model.

$$\rho\frac{\partial v}{\partial t}+\rho v\bullet\nabla v=\nabla\eta\bullet(\nabla v+\nabla v^{t})+\eta\nabla^{2}v$$
$$-(\frac{2}{3}\nabla\eta-\nabla\kappa)\bullet(\nabla\bullet v)\underline{\underline{I}}-(\frac{2}{3}\eta-\kappa)\nabla\bullet((\nabla\bullet v)\underline{\underline{I}})+\nabla p+\rho\underline{g} \quad (4)$$

Because the process is nonisothermal, heat transfer effects must be followed as well. The energy equation has a variable heat capacity, C_p, and thermal conductivity, k, both of which depend on the gas volume fraction. Heat is generated by the exothermic polymerization reaction and is used up via the evaporation of fluorinert.

$$\rho C_p\frac{\partial T}{\partial t}+\rho C_p v\bullet\nabla T=\nabla\bullet(k\nabla T)+S_{rxn}+S_{evap} \quad (5)$$

The epoxy polymerization follows condensation chemistry, where the complex kinetics can be represented by the extent of reaction ξ. The extent of reaction is calculated from the following equation which includes its time evolution, advection, and reaction kinetics.

$$\frac{\partial\xi}{\partial t}+v\bullet\nabla\xi=k^{i}e^{E_a/RT}(A+\xi^{m})(1-\xi)^{n} \quad (6)$$

Material models to populate the conservation equations were determined from experimental measurements and literature review. The choice of material model is discussed in the next section.

2.2. Material Models for Continuum Equations

In this section, we discuss the material models used to populate the continuum conservation equations discussed in the previous section. For more detailed discussion of the experiments and experimental methods that went into these material models, please see our companion, experimental SAND report [Mondy et al., 2008].

Here we summarize our variable density models, cure kinetics, complex viscosity and thermal properties.

2.2.1. Variable Density Models

In order to create a preliminary engineering model of the foam rise, we decided to simply determine a rise rate for the foam based on empirical measurements using our known mixing techniques and oven temperatures. Experiments were performed in narrow slots so that the temperature would be as uniform as possible. Typical data from a foam rise rate in a narrow slot can be seen in Figure 2, where time zero is assumed to occur after the end of the foam precursor injection. A narrow slot was used to ensure that the experiment was as isothermal as possible,

though the material must come up to temperature after being mixed at a temperature below the boiling point of fluorinert, 43°C.

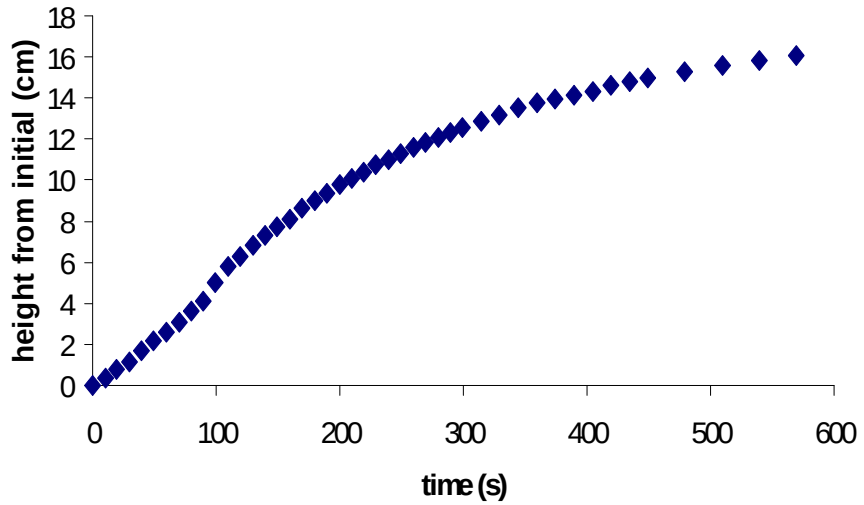

Figure 2. Height increase as a function of time for foam rise in a 65°C oven.

From Figure 3, we can see that the material comes up to oven temperature in about 100s, during which time the material is foaming. While Test A and Test B are isobaric, Test C used a pressure of 30 psia to delay foaming until the material reached the oven temperature. Unfortunately, this method of assuring an isothermal experiment leads to foam with a much higher density than the isobaric experiments. We hypothesize that the high pressure results in a two minute delay in nucleation, which is enough time for most of the air bubble to degas from the solution leaving a dearth of nucleation sites.

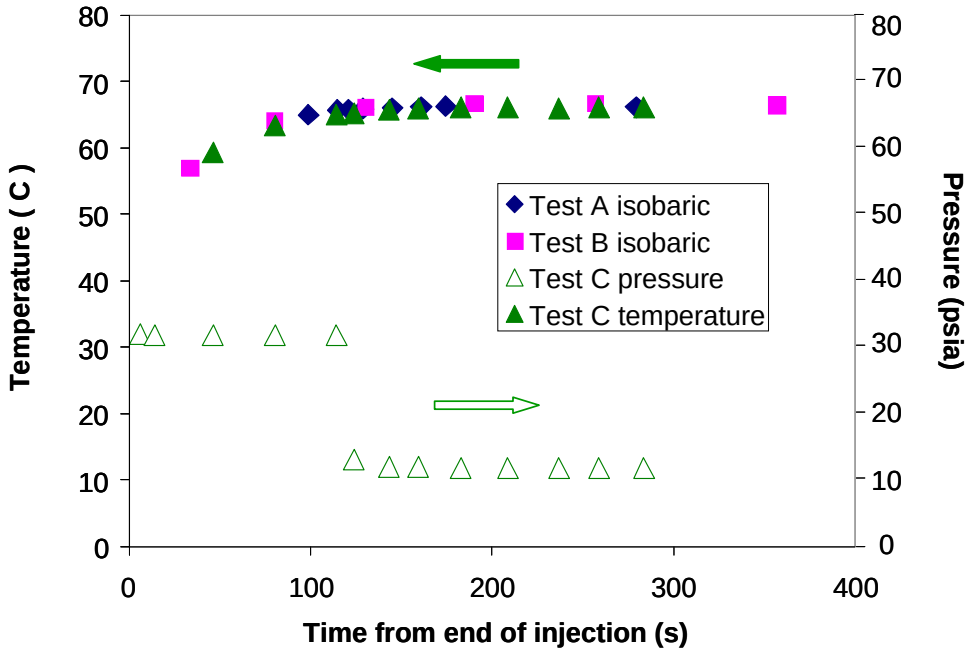

Figure 3. Temperature and pressure in foam rise rate experiments

From the foam rise rate experiments, we can determine a time-dependent density function. In order to determine this function, we must know certain parameters such as the initial volume and density, and the initial gas fraction. If we knew the initial density of the material, the height at which the material was completely injected could be calculated from the measured mass of material. Unfortunately, as the foam is expanding immediately as it enters the hot mold, it is difficult to know the density and, hence, the height at time=0. Furthermore, although we know the density of the mixture of liquid components alone is 1.14 g/cm^3 mixing incorporates a significant amount of air. Measurements of the volume of the syringe and the mass injected indicate that a typical initial density is close to 0.9 g/cm^3. This value agrees with that obtained in separate experiments in which the foam was mixed in a volume-calibrated beaker.

In order to be consistent with the mass injected, we decided to use the liquid component density, though other choices are possible. Figure 4 shows the experimental density measurement based on this choice of initial density.

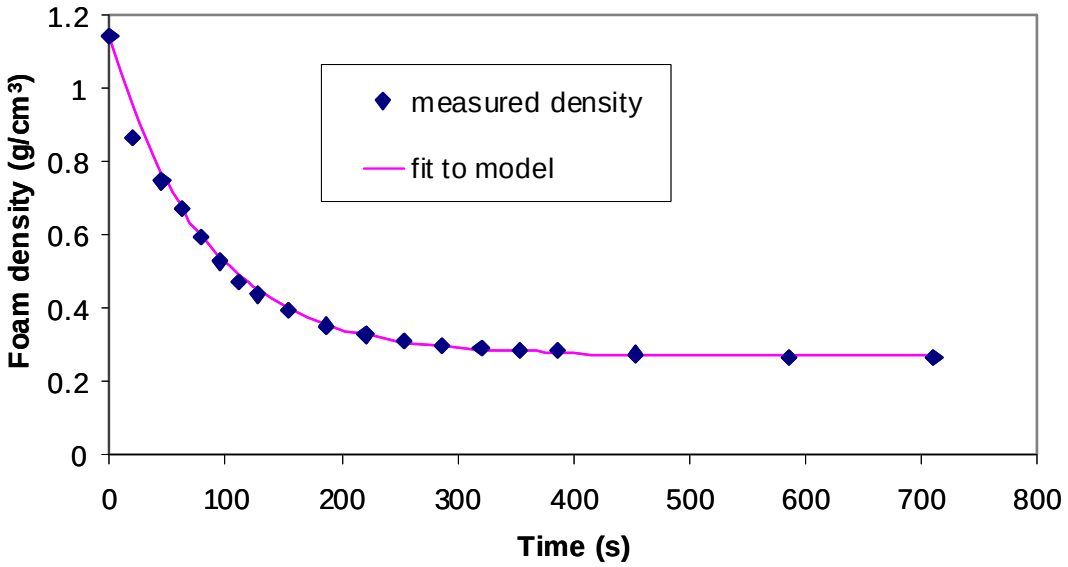

Figure 4. Density of a foam sample with time.

From the data in Figure 4, and following Seo et al, [2003], we develop a simple time-dependent density model, which assumes no spatial variations.

$$\rho = \rho_{final} + (\rho_{initial} - \rho_{final}) e^{-kt} \tag{7}$$

From the data in Figure 4, $\rho_{initial}$=1.14g/cm^3, ρ_{final}=0.27g/cm^3 and k=1/80.

Within the range of the temperatures tested, as the temperature increases, the foam rises faster. Preliminary foam rise data were taken in a similar fashion as described above, but with less control of the initial quantity of material injected into the slot and, in some cases, less temperature monitoring. However, these first measurements were taken over a higher range of oven temperatures. By considering both the preliminary data set and the subsequent more controlled data, including all the various temperatures, one can modify equation (7) to include the effects of temperature.

$$\rho = (\rho_{initial} - \rho_{final}) \exp\left[\frac{-t}{C(T)}\right] + \rho_{final}$$
$$\text{where } C(T) = \frac{A}{T} - B \tag{8}$$

Figure 5 documents the density changes at three oven temperatures, assuming that the initial density was 1.14 g/cm^3. Also shown in Figure 5 is a fit to preliminary data in the form of

equation (8) to take into account the effects of temperature and based on the nominal oven temperature only.

Here, ρ_{final} was taken to be 0.27 g/cm^3, the value typically measured in this series of experiments. T is the temperature in Kelvin. Fitting equation (8) to the data in Figure 5 resulted in values for the parameters, A and B, of 116250 K s and 274.26 s, respectively, yielding time constants for the exponential change in the foam density of 69.5 s at 65 C and 7.1 s at 140C. An actual temperature of the foam was recorded for each data point in Figure 5, so the calculated points here take into account the temperature history in this case.

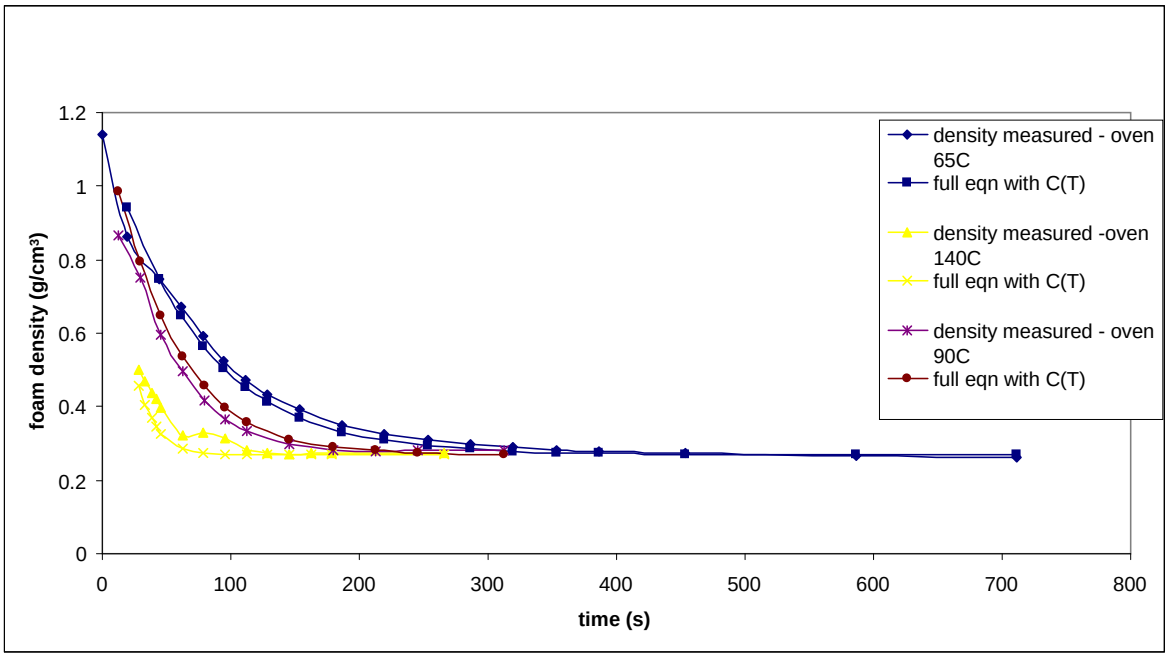

Figure 5. Foam density evolution as measured in original experiments at various temperatures, compared to equation (8) with C(*T*)=*A*/*T*-*B*.

Figure 6 shows the more recent data at both 54°C and 66°C nominal oven temperatures compared to the time-temperature-density model originally developed with the earlier data, equation (8), and calculated using the temperature history data. Also plotted are the data at the lower temperature assuming that the initial density is the liquid density ($\rho_{initial}$ = 1.14 g/cm^3). Here, the equations ignore the incorporated air and go to a final density of 0.24 g/cm^3. There is uncertainty in determining the initial density since air is incorporated both as part B is shaken to distribute the fluorinert and as part A and part B are mixed together. The liquid density is known to be 1.14 g/cm^3, but the entrained air can reduce the density to 0.9 g/cm^3 or lower.

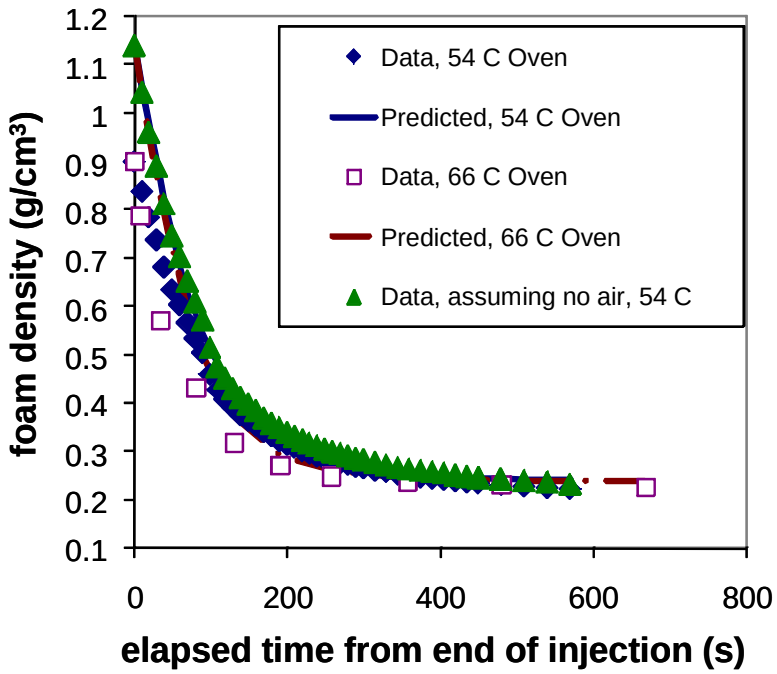

Figure 6. Foam density evolution at two temperatures. Symbols are data and lines are predicted from equation (8) assuming no initial air content ($\rho_{initial}$ = 1.14).

2.2.2. Epoxy Polymerization Model

The reaction kinetics were measured with a TA Instruments Q200 differential scanning calorimeter (DSC). The reactants were mixed at room temperature and then placed into pre-cooled test pans, which were subsequently stored at -40^{o}C to minimize reaction prior to measurement. Figure 7 shows the raw DSC data for temperatures ranging from 60^{o}C -90^{o}C thought to be representative of temperatures seen during processing and below the gel point, for heat flow with time. After the gel point, we have seen temperatures as high as 140^{o}C due to exothermic polymerization. Also shown in Figures 7, the heat flow measurements at different temperatures overlay when time, temperature shifted, indicating only one reaction mechanism. The amount of the shift reveals the activation energy, E_a, is 11 kcal/mol. The reaction is exothermic, with an average heat of reaction of 250 J/g.

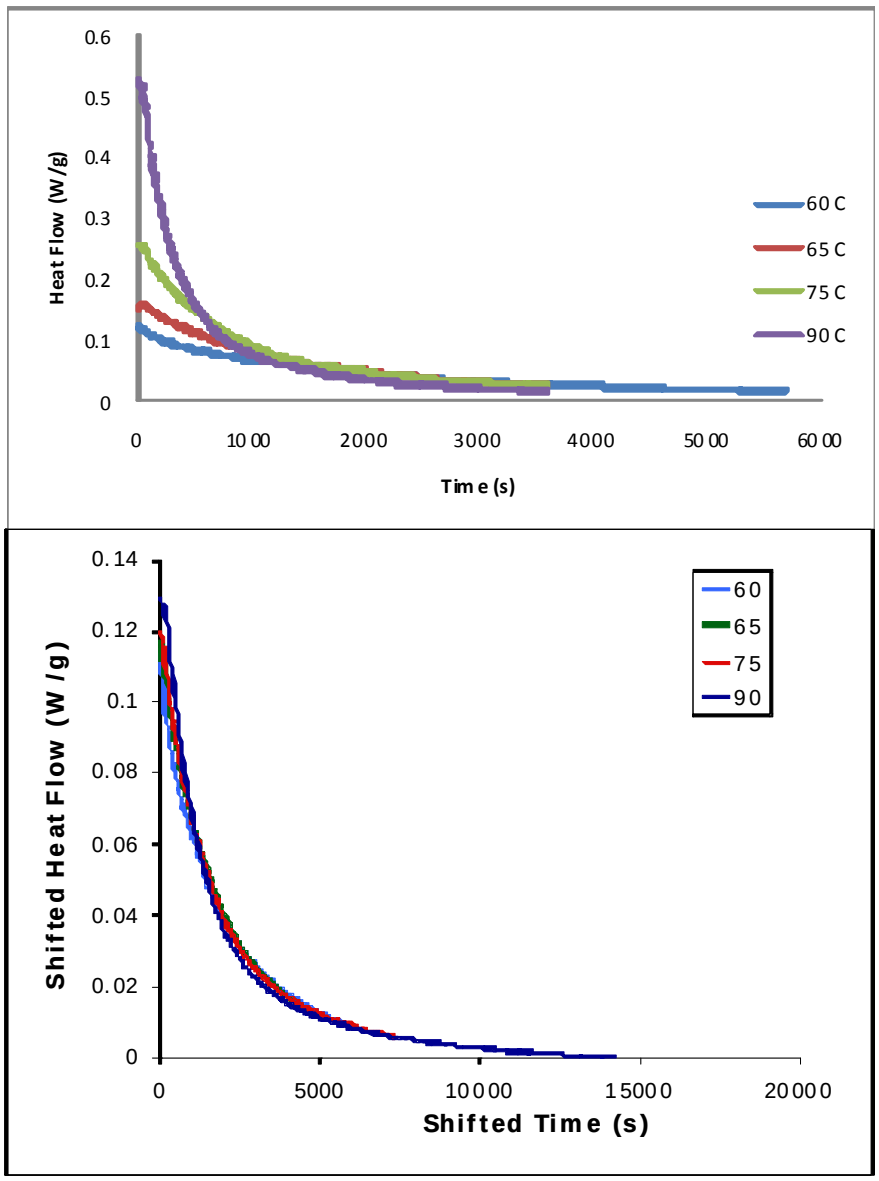

Figure 7. Raw DSC date giving heat flow with time for various isothermal experiments (top) and time-temperature shifted DSC data so all temperatures fall on the same master curve (bottom).

From the raw data, one can determine the heat of reaction associated with the polymerization, the extent of reaction, and the derivative of extent of reaction with time. With the extent of reaction and the reaction rate, in addition to the assumption that the epoxy follows condensation chemistry [May, 1988], we can populate the parameters in equation (6) and determine a kinetic rate model. Figure 8 gives a comparison on the experimental rate to the fit of the rate while figure 9 gives the extent of reaction from experiment and from the fit to the data.

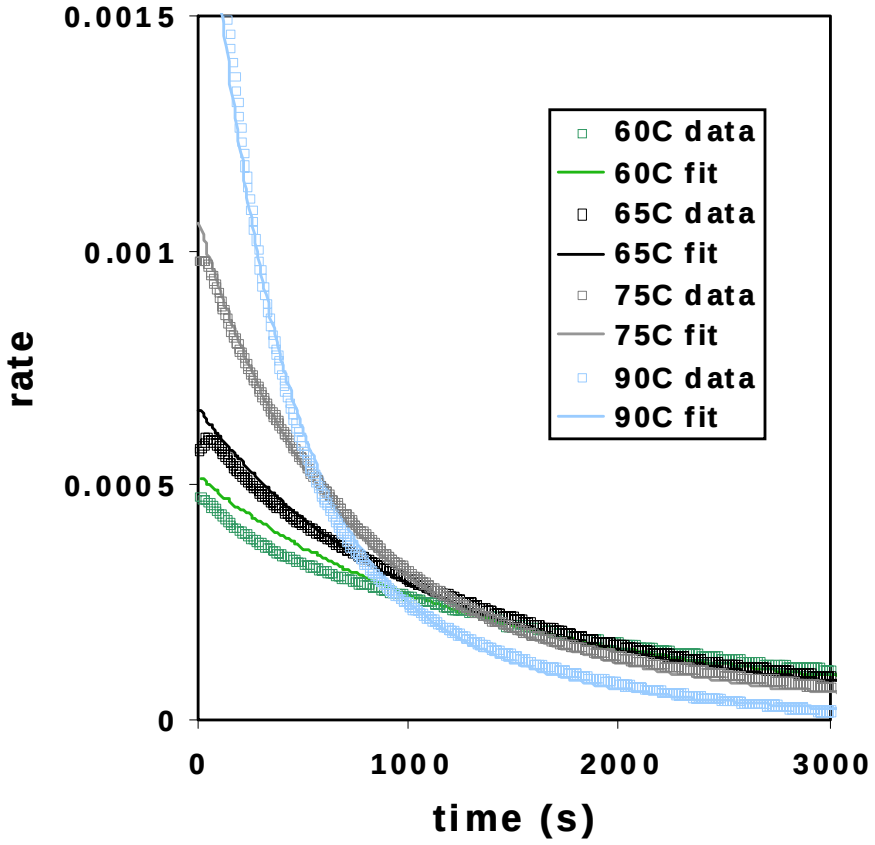

Figure 8. Experimental data compared to curve fits of the reaction rate, dζ/dt.

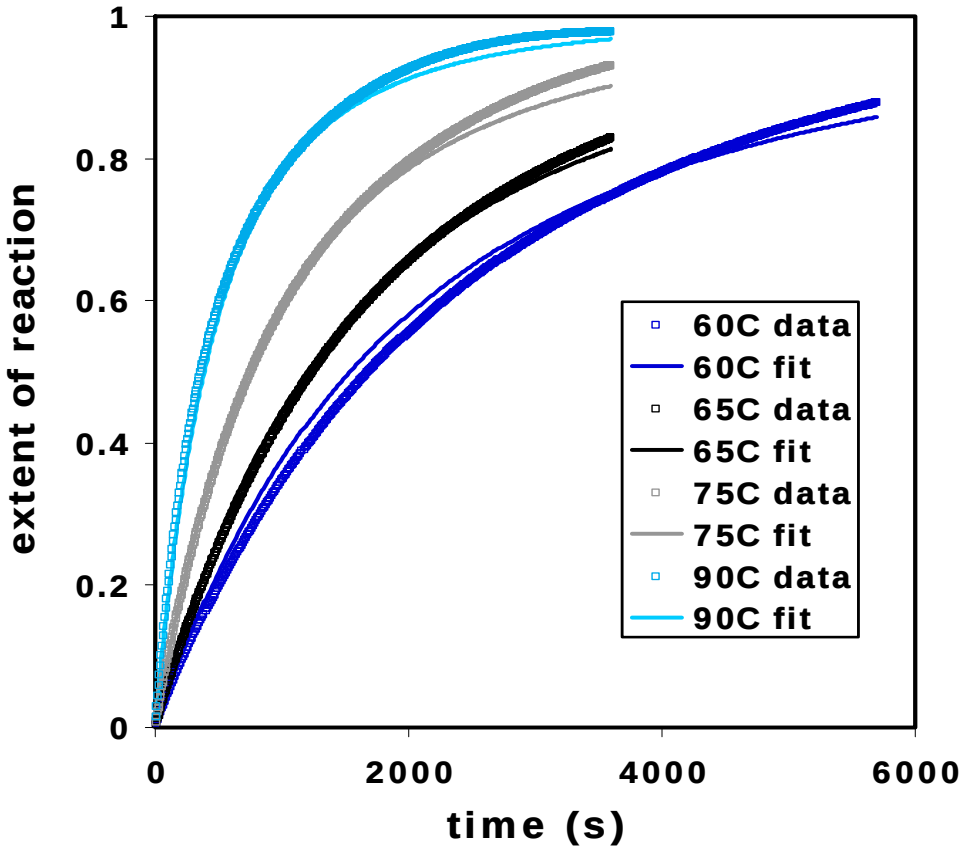

Figure 9. Experimental data compared to curve fits of the extent of reaction.

The kinetic equation for EFAR polymerization is given in the following equation with the parameters found from curve fitting substituted in. The curve fitting occurs in an Excel spreadsheet by comparing the reaction rate and extent of reaction, obtained by numerically integrating the reaction rate, to the experimental data for these quantities while varying the parameters in equation (6) using the activation energy of 11 kcal/mol determined by the time, temperature shift (Figure 7).

$$\frac{\partial \xi}{\partial t} + v \bullet \nabla \xi = 8.6 x 10^3 \frac{1}{s} e^{-\frac{11 \frac{kcal}{mol}}{RT}} (1-\xi)^{1.4} \qquad (9)$$

The parameter values for the epoxy polymerization kinetics are also summarized in Table 2.

Table 2. Epoxy Polymerization Kinetic Parameters

Epoxy Cure Parameter	Value
Rate coefficient, k^i	8.6×10^3 1/s
Activation energy, E_a	11 kcal/mol
Rate parameter, A	0
Rate exponent, m	0
Rate exponent, n	1.4
Heat of reaction, ΔH_{rxn}	250 J/g

This detailed model of the polymerization reaction can be used to determine a curing viscosity model as discussed in the following section.

2.2.3. Viscosity Models

Foam rheological properties are complex measurements to perform in a reproducible manner, since shearing the foam often changes the microstructure thereby altering the viscosity. For this reason, we decided to separate the viscosity into three parts dependent on 1) continuous phase epoxy properties, dependent on extent of reaction and temperature, 2) fluorinert emulsion effects, and 3) gas bubble volume fraction. We assume these components are multiplicative since these effects are separable and can be decoupled. This assumption is based on the suspension/emulsion literature, which has clearly shown that the effects of the continuous phase are separable from the discontinuous particle, emulsion, or gas bubble phases [Clift et al, 1978; Prud'homme and Khan, 1996].

$$\eta = \eta_{epoxy} \eta_{fluorinert} \eta_{\phi} \tag{10}$$

Rheological measurements were made for the continuous phase epoxy monomer as a function of time as it polymerized in a Rheometrics ARES rheometer with parallel plate geometry at a steady shear rate of 2 Hz. The reactants were mixed at room temperature, and the test was initiated as quickly as possible to minimize reaction prior to measurement. Various isothermal experiments were undertaken at temperature ranging from room temperature to 95^{o}C. The experimental data are shown in Figure 10.

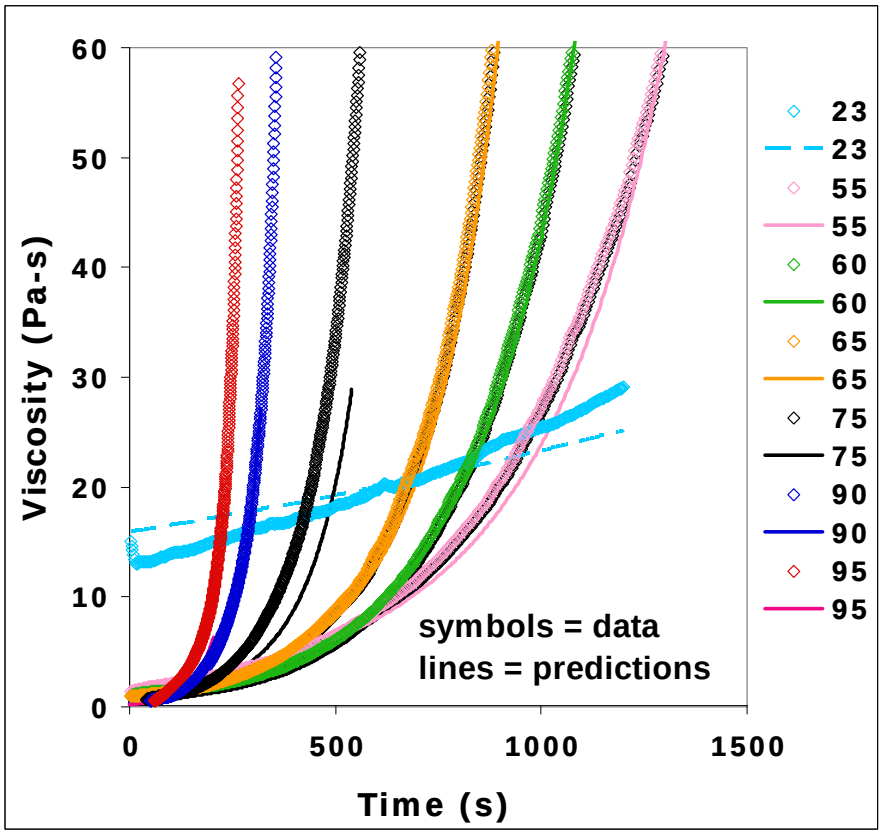

Figure 10. Viscosity prediction for the continuous phase viscosity, e.g. without foaming

For operating temperatures far above the current glass transition temperature of the reacting epoxy, the temperature dependence can be modeled accurately by an Arrhenius relationship [Ferry, 1980]. Dynamic percolation theory predicts a dependence of the Newtonian viscosity on extent of reaction with the form

$$\eta_{epoxy} = \eta_{00} \exp\left(\frac{E_a}{RT}\right)\left(\frac{\xi_c^b - \xi^b}{\xi_c^b}\right)^p \tag{11}$$

where ξ_c is the gel point, E_a is the activation energy, and η_{00} is the uncured viscosity at reference temperature T_0, and b and p are exponents for the model, with *b* being positive and *p* being negative [Martin et al., 1989]. Note, also shown in Figure 10 is the curing viscosity model fit from equation (11).

The parameters to populate this model are given in Table 3.

Table 3. Curing Epoxy Viscosity Model Parameters

Curing Epoxy Viscosity Model Parameter	Value
Uncured viscosity, at reference T, η_{00}	4.0×10^{-9} Pa s
Activation energy, E_a	13 kcal/mol
Extent of reaction at gel point, ξ_c	0.6
Curing viscosity exponent, p	-3.5
Curing viscosity exponent, b	1.0

The effect of fluorinert is harder to judge. In the foam precursor, we have an emulsion of fluorinert droplets in an epoxy continuous phase. Below the boiling point of fluorinert, the emulsion has a viscosity of about 50% higher than the neat epoxy, as seen from Figure 11.

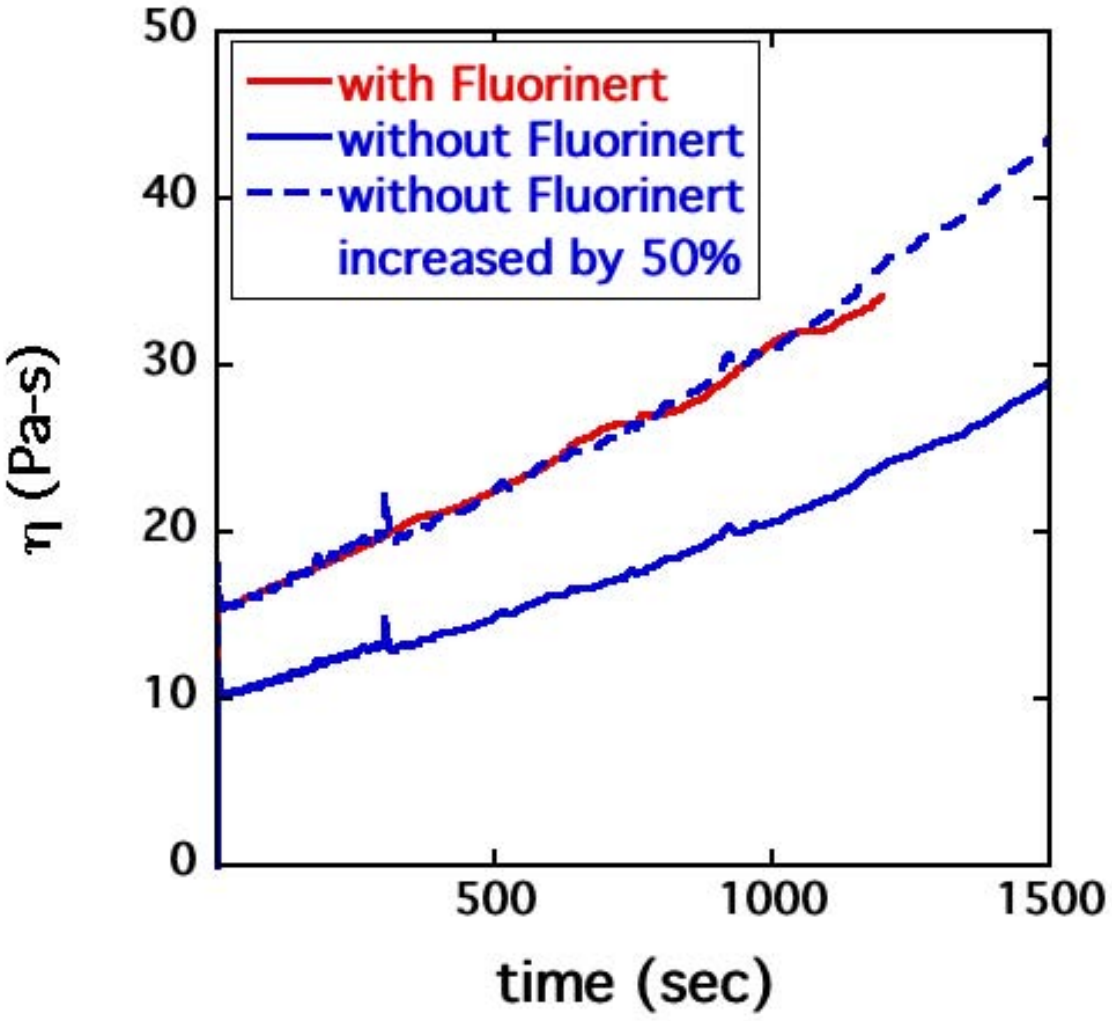

Figure 11. Effects of Fluorinert on Epoxy Viscosity. Room temperature experiments show the viscosity of the epoxy with and without the fluorinert emulsion phase.

Once the fluorinert boils and produces gas bubbles, however, it affects the viscosity differently. We have included this 50% increase in the base viscosity in the model since this will be more significant early on when the foam is expanding quickly than at later times when the viscosity is dominated by the high concentration of foam bubbles. A 50% increase is small relative to the change of viscosity compared to that from curing and gas bubbles, which can be orders of magnitude.

We expected the foam viscosity η_ϕ to be a strong function of the gas volume fraction ϕ_g and to follow the Taylor-Mooney form derived from emulsion experiments, extrapolating the discontinuous phase viscosity to zero [Prud'homme and Khan, 1995]:

$$\eta_\phi = \eta_0 \exp\left(\frac{\phi_g}{1-\phi_g}\right) \tag{12}$$

To determine if this was a satisfactory approximation to the foam effective viscosity, we tested the EFAR20 as it was foaming in a shear rheometer using the parallel plate geometry. Because the foam was expanding during the test, the foam escaped out the side of the parallel plates and, therefore, the volume of the sample did not change, but the density of the sample did. Both quantities are needed for the interpretation of the viscosity measurement. We used a temperature ramp in the rheometer that mimicked that of the free-rise experiments (Figure 12). Knowing that the viscosity of the foam would be sensitive to cell breakage, we tested a shear rate as low as possible given the resolution of the torque sensor in the rheometer. We also measured the dynamic viscosity in small amplitude oscillations to try to measure viscosity while minimizing cell breakage. Unfortunately, fluids with structure, like foams, do not follow Cox-Merz rules relating dynamic and shear viscosities. Nevertheless, the lowest shear rate measurements agreed reasonably well with the oscillatory measurements at low frequencies giving us more confidence in these low shear rate results.

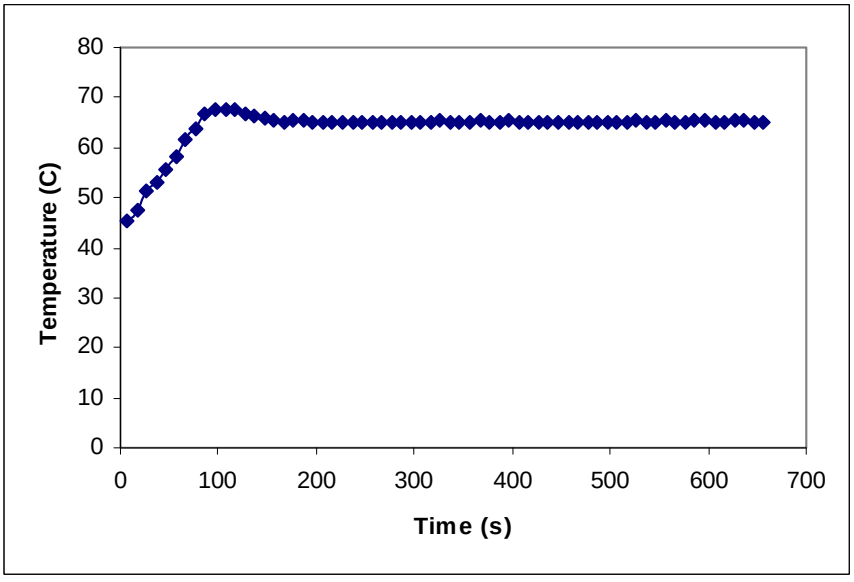

Figure 12. Temperature ramp in the free-rise experiment used to predict gas content.

Figure 13 describes the results of the rheometry tests, and shows the growth of the viscosity with time (as the foam gas fraction increases and the density decreases). The foam appears to be shear thinning, which is typical for these materials. However, as the shear rate increases, the foam structure is undoubtedly damaged, so, at a constant gas fraction, it will appear not only shear thinning but time-dependent. Nevertheless, we will interpret the low shear rate data as if the viscosity is only a function of the gas fraction, and the time dependence is only due to the increasing gas fraction. To do this the viscosity at the lowest shear rate was fit with a curve as shown on the Figure 10. The minimum viscosity was assumed to be 3.5 Pa s, estimated from the initial viscosity measured here and the corresponding predictions using equation (12).

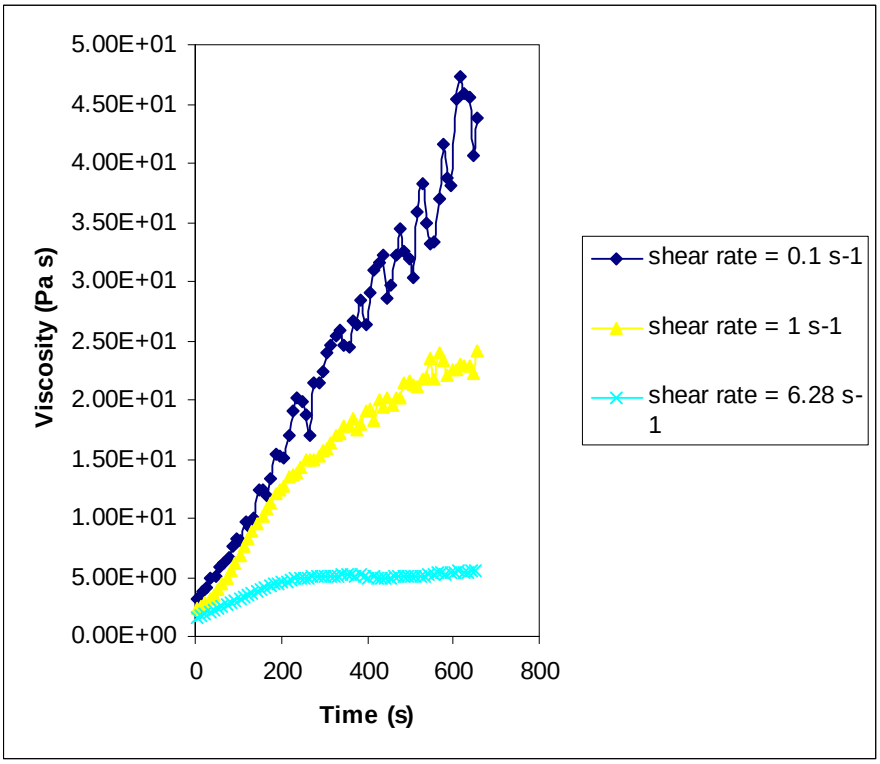

Figure 13. Parallel plate viscometry of evolving EFAR20 foam.

The gas fraction at each moment in time was estimated from the previous free-rise test, Figure 14. The measured viscosity of the foam was compared to that predicted by equation (12) and found to be adequate (Figure 14) for low shear rates. The comparison is excellent to a gas fraction greater than 0.6, while the expected limit of this equation is a gas fraction of 0.6 [Prud'homme and Khan, 1995].

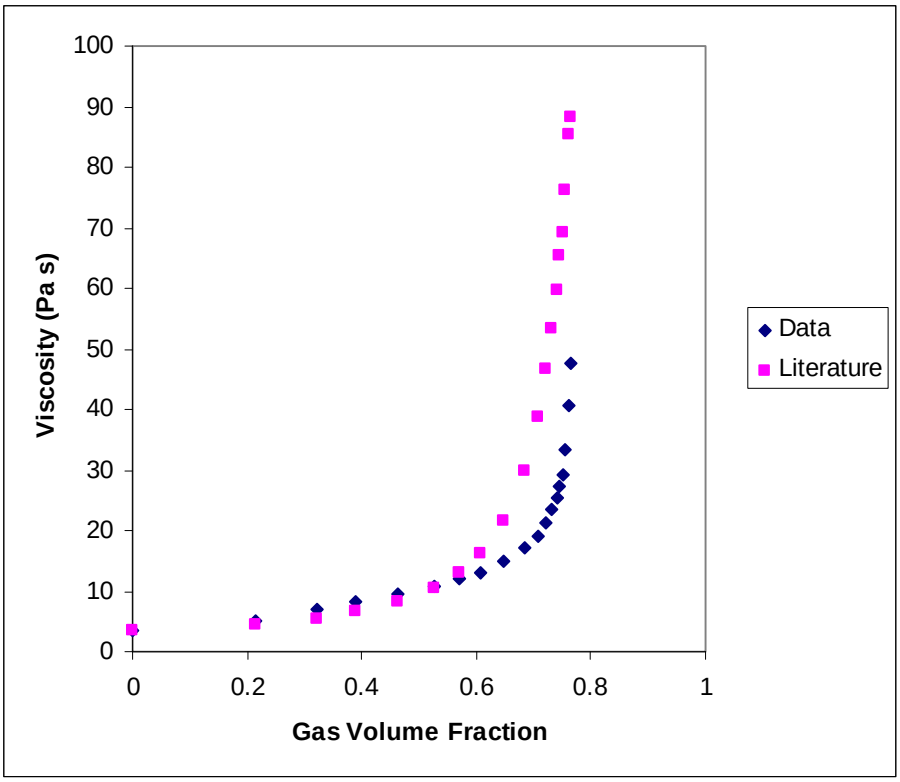

Figure 14. Foam viscosity experiments (blue) plotted with Mooney Taylor theoretical viscosity model (pink) for foam viscosity as a function of gas volume fraction.

The total shear viscosity model, including the effects of curing epoxy, fluorinert droplets, and gas bubbles is summarized in the equation below,

$$\eta_{foam} = \eta_{00} \exp\left(\frac{E_a}{RT}\right)\left(\frac{\xi_c^b - \xi^b}{\xi_c^b}\right)^p (1.5) \exp\left(\frac{\phi_g}{1-\phi_g}\right) \tag{13}$$

where in practice the effect of fluorinert is absorbed into the reference viscosity.

$$\eta_{foam} = \eta_{00} \exp\left(\frac{E_a}{RT}\right)\left(\frac{\xi_c^b - \xi^b}{\xi_c^b}\right)^p \exp\left(\frac{\phi_g}{1-\phi_g}\right) \tag{14}$$

To determine the foam stress tensor, we must understand both the shear and bulk viscosity. Following Batchelor [1967], for the simple system of an incompressible liquid with a population of compressible bubbles, expansion occurs solely through the increase in the size of the bubbles while flow occurs around each bubble. The effective expansion viscosity may be determined by equating the total dissipation as it would appear for a homogeneous fluid to the total dissipation from the ordinary shear viscosity in the liquid surrounding the bubbles. Doing these calculations yields the result

$$\kappa = \frac{4}{3}\frac{\eta_{epoxy}}{\phi_g}. \tag{15}$$

Equation (14) may be extended to non-dilute conditions, which was done by Kraynik et al [2005] who used 2D boundary integral calculations and 3D ALE free surface calculations of expansion in a Kelvin cell. Their numerical results demonstrated that the following formula,

$$\kappa = \frac{4}{3}\eta_{epoxy}\frac{(1-\phi_g)}{\phi_g}, \tag{16}$$

We should note in any case that the equation (16) results are the most pertinent because they involve processes where the divergence of velocity is large. As $\phi_g \to 0$ the liquid becomes increasingly incompressible, and the velocity becomes solenoidal ($\nabla \bullet \mathbf{v} = 0$), and bulk viscosity is no longer important.

2.2.4. Gas Production Model

The gas volume fraction, necessary for the viscosity and thermal models, can be determine from post-processing the density model and knowing some of the pure component mass fractions and densities:

$$\phi_g = \frac{1-\frac{\rho}{\rho_i}}{1-\frac{\rho_v^o}{\rho_l^o}} + \frac{\rho Y_a}{\rho_a^o} \tag{17}$$

Where ρ_i is the initial precursor foam density, ρ_v^0 is the density of pure fluorinert vapor, ρ_l^0 is the density of pure fluorinert liquid, Y_a is the mass fraction of air, and ρ_a^0 is the density of pure air. The values used for calculating the volume fraction of gas are summarized in Table 4.

Table 4. Parameters for Gas Evolution Model

Gas Evolution Parameter	Value
Density of initial foam precursor, ρ_i	1.14 g/cm^3
Density of fluorinert liquid,	1.68 g/cm^3
Density of fluorinert vapor,	.0139 g/cm^3
Mass fraction air, Y_a	0-2.6e-4

2.2.5. Thermal Properties Model

Foam heat capacity and thermal conductivity are strong functions of gas volume fraction and necessary for the energy equation, equation (5). From Gibson and Ashby [1990], the foam heat capacity can be calculated from mixture theory for a two-phase material of epoxy and gas bubbles.

$$C_p = \frac{\hat{c}_{p,l}\rho_l(1-\phi_g) + \hat{c}_{p,g}\rho_g\phi_g}{\rho} \tag{18}$$

Here $\hat{c}_{p,l}$ is the heat capacity of the continuous liquid phase and $\hat{c}_{p,g}$ is the heat capacity of the gas phase, ρ_l is the density of the liquid phase, and ρ_g is the density of the gas phase.

An upper limit for the thermal conductivity of the foam mixture is given by the following equation [Hilyard and Cunningham, 1991].

$$k = \frac{\phi_g^{2/3}k_g + k_l\left(1-\phi_g^{2/3}\right)}{\left(\phi_g^{2/3}-\phi_g\right)\frac{k_g}{k_l} + \left(1-\phi_g^{2/3}+\phi_g\right)} + k_r \tag{19}$$

This equation stems from an analysis of conduction through a solid matrix with cubic bubbles arranged in line where k_g is the gas conductivity, k_l is the liquid-phase conductivity, and k_r is the heat conduction due to radiation. In our initial implementation, we ignore any radiative contributions assuming that at an oven temperature of 65°C they will not be important. This can be tested when we undertake our thermal validation and compare this model to experimental data. Liquid phase properties were estimated by direct measurement, e.g. density, and from experience with other similar epoxy systems [Adolf, 1996]. Gas phase properties were estimated from air properties [Wikipedia, 2008].

Heat production comes from two different mechanisms: 1) the exothermic polymerization reaction of the fluorinert and

$$S_{rxn} = \Delta H_{rxn}\rho Y_e^0\frac{d\xi}{dt} \tag{20}$$

2) the endothermic evaporation of the fluorinert as it boils. The effect of fluorinert vaporization can be determined from the evolution of the density and the initial mass fractions of air and epoxy and the pure component densities.

$$S_{evap} = \frac{-\Delta\hat{H}_{evap}}{\frac{1}{\rho_v^o}-\frac{1}{\rho_l^o}}\frac{d\rho}{dt}\left(\frac{Y_e^o}{\rho_e^o} + \frac{\left(1-Y_e^o-Y_a^o\right)}{\rho_l^o} + \frac{Y_a^o}{\rho_a^o}\right) \tag{21}$$

The summary of the parameters used in the thermal models is given in Table 5. The initial mass fractions of epoxy and air can be found in Tables 1 and 4.

Table 5. Thermal Model Parameters

Thermal Model Parameters	**Value**
Heat capacity of liquid phase	2.0 J/gK
Heat capacity of gas phase	1.0 J/gK
Density of liquid phase, ρ_l	1.14 g/cm^3
Density of gas phase, ρ_g	0.001 g/cm^3
Thermal conductivity of liquid phase, k_l	0.18 W/mK
Thermal conductivity of liquid phase, k_g	0.025 W/mK
Heat of vaporization, fluorinert, $\Delta\hat{H}_{evap}$	87.1 J/g

3. NUMERICAL METHOD

The SIERRA mechanics framework is based on the finite element method and designed to solve the equations of motion for both fluid and solid mechanics and has been chosen as the nuclear weapon's tri-laboratory engineering analysis code. ARIA/SIERRA was originally developed for modeling low speed flows for applications such as manufacturing, focusing on incompressible flows. Free surface flows, using both Eulerian and Lagrangian methods, were also a focus. As part of the SIERRA consolidation process, ARIA is being broadened to include all fluid/thermal SIERRA capabilities previously available in the modules CALORE, FUEGO, and PREMO. ARIA was chosen as the development platform for foam encapsulation modeling, since it had performed well for other similar free surface flows such as epoxy encapsulation [Rao et al, 2006; Mondy et al, 2007].

3.1. Interface Tracking via the Level Set Method

As the foam expands via fluorinert nucleation and growth, the material slowly fills the mold. The location of the foam-air interface must be determined from the complex interplay of density evolution, velocity generation, viscous stress, surface tension, and gravitational effects, and it must be determined as part of the solution method. Many methods are available to determine the location of the free surface from moving mesh methods to volume of fluid methods. Here, we use an Eulerian approach based on a diffuse interface implementation of the level set method [Sethian, 1995]. Because it is an Eulerian approach it can handle geometric complexity and topological changes, such as droplet breakup, without complex remeshing steps.

The level set, $\phi(x,y,z,t)$, is a signed distance function of space and time. The magnitude of the level set function is the shortest distance from *x* to any point on the free surface, where the free surface is defined by the level set zero. The sign of level set is used to indicate whether the point x lies inside the material. It should be noted that φ should scale as a distance function, that is, the magnitude of its gradient is unity.

$$\left|\nabla\phi\right| = 1 \tag{22}$$

This level set representation of the interface presents numerous advantages. The location of the interfacial curve can be determined exactly from interpolation of the finite element shape functions. In addition, the level set representing function provides immediate information about the normal, *n*, and curvature, $\mathcal{H}$, of the interfacial surface via these relations:

$$\begin{aligned} n_{ls} &= \nabla\phi \\ \mathcal{H} &= -\frac{\nabla^2\phi}{\left|\nabla\phi\right|} \end{aligned} \tag{23}$$

Because the level set zero,

$$\phi(x,y,z,t) = 0\,, \tag{24}$$

is a material surface, it advects with the fluid velocity whereas elsewhere it is unclear how the level set equation evolves. For this reason, we use an advection equation for the entire level set function.

$$\frac{\partial \phi}{\partial t} + u \cdot \nabla \phi = 0 \tag{25}$$

Because advection is only truly applicable at the interface, equation (24) distorts the distance function. This necessitates the use of a renormalizing algorithm, which must be run periodically to correct the distance function. Here, we utilize a constrained redistancing algorithm that attempts to redistance the level set so that the volume of each phase remains unchanged. Details of the renormalization algorithm can be found below.

3.1.1. Property Evaluation

Notationally, we denote these two sides of the interface as “phase A” and “phase B” and often use “A” and “B” as subscripts on mathematical quantities that are specific to a phase. By convention, we associate phase A with the negative level set field. This can be seen from an example ARIA input file (Appendix A), where we have properties associated with phase A, phase B, and multiphase properties that apply to both phases.

The level set method uses continuous equations for both the foam and gas phase, but modulates the material properties based on the level set function. This property modulation occurs via a numerical Heaviside function defined for phase A and B, where the Heaviside functions sum to one.

$$H_A(\phi) + H_B(\phi) = 1 \tag{26}$$

In the diffuse interface approach, the Heaviside function is regularized so that there is a smooth transition from one phase to another. In this work, we use

$$H_B(\phi) = \frac{1}{2}\left(1 + \frac{\phi}{2} + \frac{\sin\left(\frac{\pi\phi}{\alpha}\right)}{\pi}\right), \ -\alpha < \phi < \alpha \tag{27}$$

where α is defined as the half of the width of the diffuse interface, which is usually taken as about six elements across. Here H_B is zero in phase A, 1 in phase B, and follows equation (27) in the diffuse region.

Another useful function related to the Heaviside function is the regularized Dirac delta function, which is defined as

$$\delta_\alpha(\phi) = \frac{dH(\phi)}{d\phi} = \frac{|\phi|}{2\alpha}\left(1 + \cos\left(\frac{\pi\phi}{\alpha}\right)\right), \ -\alpha < \phi < \alpha \tag{28}$$

where $\delta_\alpha(\phi)$ is large in the diffuse interface zone, and zero elsewhere.

The continuity equation, (1) can be rewritten including the Heaviside function to modulate the density from foam to gas phase.

$$(H_A\rho_A + H_B\rho_B)\nabla\bullet v = -H_A(\frac{\partial\rho_A}{\partial t} + v\bullet\nabla\rho_A) - H_B(\frac{\partial\rho_B}{\partial t} + v\bullet\nabla\rho_B) \tag{29}$$

Assuming here that the foam phase is designated A and the gas phase designated B, we assume the foam has a variable density and ignore any density variations in the air phase since there variations will be orders of magnitude smaller. This yields:

$$(H_A\rho_A + H_B\rho_B)\nabla\bullet v = -H_A(\frac{\partial\rho_A}{\partial t} + v\bullet\nabla\rho_A)\,. \tag{30}$$

The momentum equation, (2) including property modulation becomes,

$$(H_A\rho_A + H_B\rho_B)(\frac{\partial v}{\partial t} + v\bullet\nabla v) = H_A\nabla\bullet\underline{\underline{\pi}}_A + H_B\nabla\bullet\underline{\underline{\pi}}_B + (H_A\rho_A + H_B\rho_B)\underline{g} \tag{31}$$

where the phase defined stress tensor is more complex for the foam, while the gas is assumed to have a constant viscosity and be incompressible.

$$\begin{aligned}\underline{\underline{\pi}}_A &= \underline{\underline{\tau}}_A - p\underline{\underline{I}}\\ \underline{\underline{\pi}}_B &= \underline{\underline{\tau}}_B - p\underline{\underline{I}}\end{aligned} \tag{32}$$

The energy equation (5), becomes,

$$(H_A\rho_A C_{pA} + H_B\rho_B C_{pB})(\frac{\partial T}{\partial t} + v\bullet\nabla T) = \nabla\bullet((H_A k_A + H_B k_B)\nabla T) + H_A S_{rxn} + H_A S_{evap} \tag{33}$$

with the heat source terms only occurring in the foam phase.

The epoxy polymerization equation is also phase dependent and only has a nonzero right-hand-side source term in the foam phase.

$$\frac{\partial\xi}{\partial t} + v\bullet\nabla\xi = H_A k^i e^{E_a/RT}(A + \xi^m)(1-\xi)^n \tag{34}$$

3.1.2. Surface Tension

Mold filling flows are often highly influenced by the capillary dynamics at the fluid-gas interface, though it is unclear how important surface forces are to foam self-expansion flows. Due to the implicit tracking of this interface, special care must be taken to enforce the capillary

boundary condition. The general form of the capillary boundary condition for constant surface tension, σ, is

$$\underline{n}\bullet(\underline{\underline{\pi}}_A - \underline{\underline{\pi}}_B) = -2\mathcal{H}\sigma\underline{n} \tag{35}$$

Integrated boundary conditions, such as equation (35), can be applied with the level set method using several techniques [Jacqmin, 1995], only one of which we discuss here. To evaluate equation (35), we must construct a local mean curvature $\mathcal{H}$. The local mean curvature is defined as

$$-\mathcal{H} = \nabla\bullet n \approx \nabla\bullet n_{ls} = \frac{\nabla^2\phi}{|\nabla\phi|} \tag{36}$$

Hence, the curvature is related to the second derivative of ϕ. Since we use the standard finite element discretization for ϕ, it is only C^0 continuous and can only be differentiated once. Thus, in this work, we solve for an approximation to the curvature, $\mathcal{H}_p$, using a least squares, lumped mass projection that is integrated by parts, creating a boundary term in addition to the volume integral.

$$\int_V N_i\mathcal{H}_p dV = -\int_V N_i(\nabla\bullet n_{ls})dV = -\int_V \nabla N_i\bullet n_{ls}dV + \int_S N_i n\bullet n_{ls}dS \tag{37}$$

Here, n is the unit normal along the exterior boundary of the domain. In order to enforce a prescribed contact angle, $\theta = \cos^{-1}(n\bullet n_{ls})$ one would include the boundary term in (37) and replace $n\bullet n_{ls}$ with cos θ. For the application described here, we weakly impose θ = 90 and so this boundary contribution is not included.

Since there is no explicit interface at ϕ = 0, we utilize the diffuse Dirac delta function and the volumetric projection for the curvature $\mathcal{H}_p$ to implement the capillary boundary condition, equation (34), viz.,

$$n\bullet(\underline{\underline{\pi}}_B - \underline{\underline{\pi}}_A) = 2\sigma\delta_\alpha(\phi)\mathcal{H}_p\underline{n}_{ls} \tag{38}$$

following the continuous surface force literature [see for instance, Brackbill et al., 1992].

3.1.3. Redistancing Algorithm

One distinct aspect of the level set method is that while the level set function might initially have the smooth properties of a distance function, this is not a necessarily preserved by the evolution scheme. It is almost certainly the case that as evolution proceeds it will deviate away from a pure distance function. Indeed sharp gradients might occur at some points in the flow, while very shallow gradients occur in others. The former is bad because it leads to inaccuracies in the integration of equation (25); that latter is bad because it widens inappropriately the thickness of

the interfacial zone. A necessary aspect of any level set method, therefore, is a periodic need to redistance or renormalize the level set function back to a distance function. Our decision to renormalize is based upon monitoring the average gradient magnitude of the level function over the interfacial zone. In general, this average gradient is only allowed to vary between 1.25 and 0.75. Outside of this range it will trigger a redistancing procedure.

There are several methods by which this can be done. Sussman and Fatemi [1999] described a redistancing step based upon a separate evolution of the level set field subject to a mass conserving constraint. We have implemented a more algorithmic approach. The elements that contain the interface can be quickly identified as those whose nodal level set values have differing signs. On these elements, a piecewise linear representation of the interface is constructed. For each node i in the mesh, it is possible to find a minimum distance D_i to this set of facets. Renormalization of the level set nodal unknown is made by the simple assignment:

$$\phi_j^* = sign(\phi_j^0) D_j \tag{39}$$

where ϕ_j^0 is the value of the level set function prior to renormalization. Given a sufficient density of facets, this procedure will yield good results as well as being fast and robust. However, it does present the potential for slight, systemic motion of the zero level set contour and a consequent loss of mass. This is especially a problem for lower order (trilinear) interpolation of the level set function. This can be avoided however by introducing a volume constraint. This is accomplished by finding a small change to the distance function, ε, such that the initial and final volumes are the same. The distance function at the end of the renormalization is thus given by,

$$\phi_j = \phi_j^* + \varepsilon \tag{40}$$

where ε is found by solving the equation,

$$\int_V H_A^0\left(\phi^* + \varepsilon\right) dV = \int_V H_A^0\left(\phi^0\right) dV \tag{41}$$

The superscript on the Heaviside is used here to denote that this is the sharp Heaviside function, that is unity where its argument is positive and zero else where. The left hand side of this equation is the volume of the phase A following the renormalization and the right hand side is the volume before renormalization. By using this adjustment to the nearest point distance, we guarantee that the volume of the phase A after renormalization will be the same as it was before renormalization.

3.2. Finite Element Discretization

The phase modulated equations of motion, conservation of energy, and kinetics equation (29-34) together with the level set equation (25), are discretized with the well-known Galerkin finite element method. For details of the finite element method see for instance Hughes [2000]. The unknowns of interest are the velocity vector, pressure, temperature, extent of reaction, and level set. These fields are approximated with finite element basis function, $N_i(x,y,z)$, and nodal

variables, e.g. u_i, v_i, w_i, T_i, ξ_i, ϕ_i. Bilinear Lagrangian, C_0-continuous, basis functions are used for all variables. The velocity vector would be expressed in the following manner,

$$u = \sum_{i=1}^{n} u_i N_i(x,y,z),\ v = \sum_{i=1}^{n} v_i N_i(x,y,z),\ w = \sum_{i=1}^{n} w_i N_i(x,y,z) \tag{42}$$

and pressure, temperature, extent of reaction, and level set are

$$p = \sum_{i=1}^{n} p_i N_i(x,y,z),\ T = \sum_{i=1}^{n} T_i N_i(x,y,z),\ \xi = \sum_{i=1}^{n} \xi_i N_i(x,y,z),\ \phi = \sum_{i=1}^{n} \phi_i N_i(x,y,z)\,. \tag{43}$$

The approximate variables are substituted in the conservation equation, multiplied by a weighting function, and integrated over the domain. In the Galerkin finite element method, the weight function is chosen to be the bilinear shape functions themselves. Any second derivatives, such as the divergence of the stress tensor and the heat flux, are integrated by parts, to improve the accuracy of the discretization as shown below. The integration by parts on the momentum and energy equations kick out surface terms that serve as natural boundary conditions if no other conditions are applied at the domain boundaries.

$$R_i^{continuity} = \int_V N_i \left[\rho \nabla \bullet v + \left(\frac{\partial \rho}{\partial t} + v \bullet \nabla \rho\right) \right] dV = 0 \tag{44}$$

$$R_i^{momentum} = \int_V \left(N_i\left(\rho \frac{\partial v}{\partial t} + \rho v \bullet \nabla v - \rho \underline{g}\right) - \nabla(\underline{e} N_i) : \underline{\underline{\pi}}\right) dV + \int_S n \bullet \underline{\underline{\pi}} \bullet \underline{e} N_i dS = 0 \tag{45}$$

$$R_i^{energy} = \int_V \left(N_i\left(\rho C_p \frac{\partial T}{\partial t} + \rho C_p v \bullet \nabla T - S_{rxn} - S_{evap}\right) - \nabla N_i \bullet k \nabla T\right) dV + \int_S n \bullet k \nabla T dS = 0 \tag{46}$$

$$R_i^{cure} = \int_V N_i\left(\frac{\partial \xi}{\partial t} + v \bullet \nabla \xi - k^i e^{E_a/RT}(A + \xi^m)(1-\xi)^n\right) dV = 0 \tag{47}$$

$$R_i^{levelset} = \int_V N_i\left(\frac{\partial \phi}{\partial t} + v \bullet \nabla \phi\right) dV = 0 \tag{48}$$

Time derivative are discretized using a first-order backward Euler finite difference method. The resulting weighted residual equations are integrated numerically using Gaussian quadrature.

There exists an inf-sup condition constraining the pressure space to be one order lower than the velocity space, which is termed the LBB condition [Hughes, 2000]. For 2D problems, we routinely use quadratic velocity basis functions coupled to linear or bilinear pressure interpolation. This is a computational intensive pair, because high order Gaussian quadrature must be used and nodal summations per element increase from four for bilinear to nine for biquadratic. In addition, due to the fact that the continuity equation, (1), is the equation for the

pressure unknown and contains no pressure, a saddle point problem is created when the discretized matrices are formed. For this reason, direct Gaussian elimination is needed to invert the matrix and solve the unknowns of interest. In 2D, this is all feasible since bandwidths are small. In 3D, we cannot afford the cost of either high order elements or direct solvers. Therefore, to allow us to use equal order interpolation and circumvent the LBB condition, we must use a stabilization method, an extra benefit of which is allowing us to use Krylov-based interative solvers in place of direct solution methods. Details of the stabilization method used here are discussed in the following section.

3.2.1. Pressure Stabilization

In order to circumvent the LBB condition, one must either use a compatible pair of finite element basis functions or add stabilization terms to relieve the mathematical restrictions. Many types of stabilization exist, the most common being Galerkin least squares (GLS) or pressure stabilized Petrov-Galerkin (PSPG) popularized by Hughes and coworkers [Hughes, 2000]. These methods work well for moderate to high Reynolds' number applications, but have issues at the Stokes limit. Other complexities arise relating to the scaling of the stabilization term and many papers have been written trying to determine the best scaling coefficient.

Here, we employ the stabilization technique developed by Dohrmann and Bochev [2004], which was developed for the Stokes problem and is termed pressure stabilized pressure projection (PSPP). In addition to being computationally efficient and easy to implement, this stabilization method is also easy to use since it requires no special treatment for boundary conditions. Numerically, this method simply requires an additional volume integral term in the weak form of the continuity equation. Scaling of the stabilization term relies on dimensional analysis and does not contain an explicit mesh scaling. The first term is the standard Galerkin continuity weighted residual, while the second term is the stabilization.

$$
\begin{aligned}
R_i^{continuity} &= \int_V N_i \left[\rho \nabla \bullet v + \left(\frac{\partial \rho}{\partial t} + v \bullet \nabla \rho \right) \right] dV + \sum_{elem}^{M} \tau_{pspp} (N_i - \pi N_i)(p - \pi p) dV \\
\pi p &= \int_{V_e} p dV \Big/ \int_{V_e} dV
\end{aligned}
\tag{49}
$$

For multiphase level set problems, there exist a pressure jump and the fluid-gas interface due to surface forces. Because we are applying a stabilization method developed for a single phase problem to a multiphase one, we have made a slight modification to stabilization method that allows us to capture the pressure jump at the fluid-gas interface. Instead of stabilizing on the pressure variable, we stabilize on the product of the time derivative of the pressure and the time step size as shown below.

$$R_i^c = \int_V N_i \left[\nabla \bullet v + \frac{1}{\rho}\left(\frac{\partial \rho}{\partial t} + v \bullet \nabla \rho\right) \right] dV + \sum_{elem}^{M} \tau_{pspp} \Delta t (N_i - \pi N_i)\left(\frac{\partial p}{\partial t} - \pi \frac{\partial p}{\partial t}\right) dV$$
$$\pi \frac{\partial p}{\partial t} = \int_{V_e} \frac{\partial p}{\partial t} dV \Big/ \int_{V_e} dV \tag{50}$$

This method has shown itself superior for mass conservation in level set problems compared to equation (48) either with a single or dual pressure formulation.

3.2.2. Taylor-Galerkin Upwinding for Level Set Equation

The level set equation is purely hyperbolic, while the Galerkin finite element works most effectively on elliptic differential equations. For that reason, we apply a Taylor-Galerkin upwinding term to the level set equation to help it behave well away from the interface. In the Taylor-Galerkin weighted residual of the level set advection equation (25), the first two terms are the standard weighted residual advection operator, while the third term is the upwinding term.

$$\int_V N_i \frac{(\phi_i^{n+1} - \phi_i^n)}{\Delta t} dV = -\int_V N_i (v^{n+1} \bullet \nabla \phi_i^{n+1}) dV - \frac{\Delta t}{2} \int_V (v^n \bullet \nabla N_i)(v^n \bullet \nabla \phi_i^n) dV \tag{51}$$

Since the magnitude of the gradient of the level set function is by definition a nearly constant value, the Taylor-Galerkin contribution is relatively small. This points to one of the advantages of using a smooth function to represent the interface location.

3.2.3. Streamline Upwind Petrov-Galerkin for the Momentum Equation

As we move from the Stokes' regime to a moderate Reynolds' number, we find that the differential equation changes from purely elliptic to hyperbolic-elliptic and some stabilization method is useful for the momentum equation. As our gas is displaced by foam in the mold, the gas phase velocity increases, resulting in a Reynolds' number in the 10-100 range. This can trigger numerical issues such as oscillations in the solution. We have found that adding some streamline upwinding Petrov-Galerkin (SUPG) improves the performance of the momentum equations and reduces oscillations in the pressure and velocity. The SUPG method involves a modified weight function, W_i, consisting of the shape function plus the velocity dotted into the gradient of the weight function multiplied by a scaling factor based on element size.

$$W_i = N_i + \sum_{elem}^{M} \tau_{supg}(h_{elem}) \frac{v \bullet \nabla N_i}{|v|} \tag{52}$$

The weight function is applied only to the terms that are not integrated by parts, since a gradient cannot be integrated by parts. This makes the SUPG method an inconsistent method for low to moderate Reynolds' numbers where the total stress dominates the residual. However as the element size goes to zero it will asymptote to the correct partial differential equation.

$$R_i^{momsupg} = \int_V (W_i(\rho\frac{\partial v}{\partial t} + \rho v \bullet \nabla v - \rho \underline{g}) - \nabla(\underline{e}N_i) : \underline{\underline{\sigma}})dV + \int_S n \bullet \underline{\underline{\sigma}} \bullet \underline{e}N_i dS = 0 \quad (53)$$

The SUPG primarily offers upwinding in the steamwise direction.

3.3. Matrix Equations and Krylov-Based Iterative Solvers

Once the residual equations are integrated, we obtain a set of nonlinear algebraic equations on each element that must be gathered into a global matrix, and solved for the nodal unknowns. We solve the level set equation in a separate matrix from the rest of the unknowns, since decoupling the equations seems to make the method more robust and improve convergence.

$$\begin{aligned} F(v, p, T, \xi) &= 0 \\ G(\phi) &= 0 \end{aligned} \quad (54)$$

This system of equations is linearized and solved with the Newton-Raphson method, where F has been expanded in a Taylor series about the kth iterate:

$$\begin{aligned} &F(v, p, T, \xi) + \frac{\partial F}{\partial v}|^k (v^{k+1} - v^k) + \frac{\partial F}{\partial p}|^k (p^{k+1} - p^k) + \frac{\partial F}{\partial T}|^k (T^{k+1} - T^k) + \frac{\partial F}{\partial \xi}|^k (\xi^{k+1} - \xi^k) = 0 \\ &\frac{\partial G}{\partial \phi}|^k (\phi^{k+1} - \phi^k) = 0 \end{aligned} \quad . (55)$$

All the terms involving the kth iterate are gathered and placed on the right-hand-side of the equation. This results in a matrix equation of the form:

$$K(x^k)\Delta x^{k+1} = f \quad (56)$$

where K is an analytical Jacobian matrix and we solve for the unknown update from the last iteration, Δx^{k+1}. We have one matrix equation for the bulk fluid unknowns and one for the level set unknowns.

As discussed above, in 2D we generally solve the fluid matrix equations with direct Gaussian elimination, but in 3D this becomes impossible since the matrix bandwidth is so much larger. Instead, we use Krylov-based iterative solvers from Trilinos, an open source parallel solver library developed at Sandia National Laboratories [Trilinos, 2008]. The stabilization of the continuity equation, equation (46), has an additional benefit besides allowing equal order interpolants: it also adds a pressure term to the pressure equation, improving the diagonal dominance of the matrix equations (56), and reducing the matrix condition number. Needless to say, it would be near impossible to use iterative solvers on equation (56) without stabilization. The discretized level set matrix equations, on the other hand, tend to be well-behaved as long as the velocity is well-behaved.

ARIA/SIERRA can be run in serial or parallel, where the parallel implementation is based on MPI. For the problem discussed here, we focus on parallel solution run on 30-68 processors on the Engineering Sciences LAN and Sandia National Laboratories capacity computing platform, Thunderbird. Though processors are slower on Thunderbird than the LAN, run times are faster since communication speeds between processors are faster than on the LAN. Each run on Thunderbird would take about 16-25hrs, depending on the physics models used.

Parallel iterative solvers always require preconditioning. Some of the preconditioners suitable for level set foaming problems are ILU and ILUT, with fill factors of 1-3. Note, as we increase the fill factors, the time to solve the matrix equations increases sharply. (Others have had luck with ML preconditioning, though we have not tested it on the foam expansion problem yet.) Solvers used here range from BiCGStab to GMRES. Generally, when the solvers are having difficulty, we resort to ILUT(3)/GMRES, which is the most robust, and most expensive, choice.

3.4. Geometry, Mesh, Initial Conditions, and Boundary Conditions

KCP, the production encapsulation facility for the W76-1 firing sets, has developed a quality assurance tool that is pictured in Figure 15 [Mahoney, 2007]. This geometry is an unlimited release version of the firing set geometry called the firing set quality assurance (QA) fixture. Mahoney has also developed a similar QA fixture for the AFS foaming. These fixtures can be used to test new foaming processing techniques and protocols without wasting expensive WR components.

For the QA fixture, a complex channel is machined in an aluminum block and a clear acrylic cover is held on the front face with screws. To monitor the quality of the foam during an encapsulation process, this mold is filled with the foam encapsulant and monitored to make sure that it fills the part. Filling is through injection ports in the left hand corner of either the inner cup shape or the outer serpentine shape as shown in Figure 15. We took this mold, copied it, and instrumented it with four thermocouples. The first, thermocouple TC101, is in the injection port machined through the back wall. It is not quite into the main reservoir. Thermocouples TC102 and TC103, are in the foam channel as pictured, about halfway between the front face of the back wall and the inner surface of the front cover and about halfway across the channel width. The fourth, TC104, is within the mold itself, centered in the aluminum block about 0.16 cm (1/16 in.) from the inner face of the back wall, and so never touches foam.

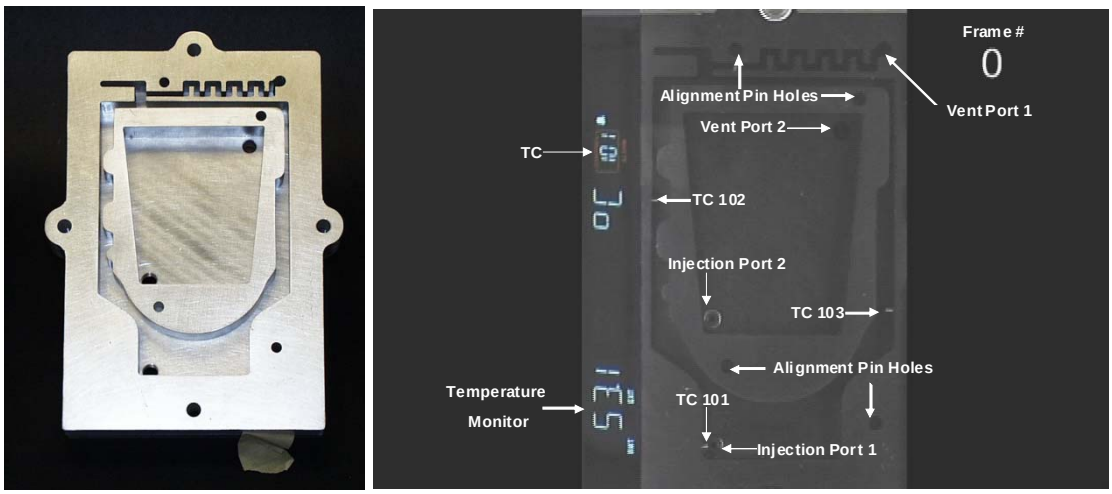

Figure 15. Kansas City mold (left), as seen in the videos and annotated (right).

The outer mold, consisting of narrow sections and serpentine routes for the foam to penetrate, was used for model validation studies. The inner cup was filled in the initial trials, but later ignored.

3.4.1. Finite Element Mesh and Boundary Conditions

We received a mesh based on this QA fixture geometry from KCP [Mahoney, 2007]. This mesh was used to test our foam expansion model and is given in Figure 16.

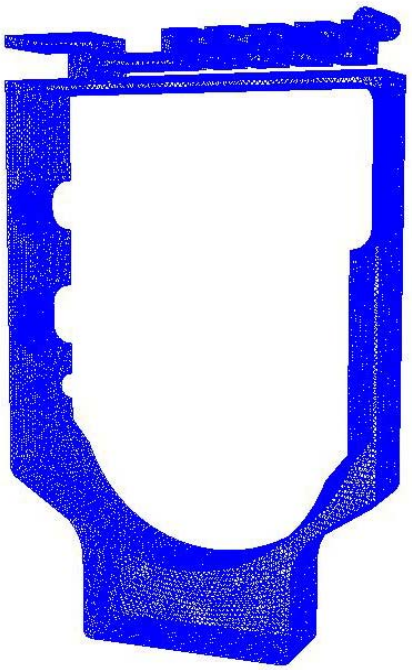

Figure 16: Tetrahedral Mesh for QA Test Fixture

The mesh consists of 31,627 nodes and 129,272 tetrahedral elements. The QA fixture contains flow restrictions and sharp corners making it a good test of real materials and processes as well as numerical algorithms. We have tested the method discussed in the previous sections on this geometry.

The mesh includes only the regions of fluid flow and not the aluminum mold, which could be included for heat transfer calculations. Instead of adding the mold for nonisothermal runs, we instead use a heat transfer coefficient at all solid boundaries of the form:

$$q = h(T - T_{oven}). \tag{57}$$

We estimate a heat transfer coefficient of roughly h=2.5 J/Kcm^2 s. We could have also directly applied the mold temperature from experiment to the surface of the mold, since this seemed to hold constant at the oven temperature, e.g. a Dirichlet condition of

$$T = T_{oven} \tag{58}$$

may also be appropriate.

Boundary conditions are also necessary for the velocity equation to indicate solid surfaces from which the foam cannot leak through. Foam is notorious for slipping, so here we apply a Navier-slip condition in the tangential direction and a no penetration condition in the normal direction,

$$\begin{aligned} n \bullet \tau \bullet t &= \frac{1}{\beta}(v - v_s) \bullet t \\ v \bullet n &= 0 \end{aligned} \tag{59}$$

where β is the Navier slip coefficient, estimated from experiments to range from 0.01-0.001, t is the surface tangent, and v_s is the velocity of the surface, here taken to be zero. In our simulations, we have found that β smaller than 0.01 creates numerical problems such as entrained air layers. We are currently investigating material dependent slip models, since the gas phase is thought to slip more than the foam phase, a reality not incorporated in the standard Navier slip condition with a constant β.

Because these boundary conditions are stated in a normal/tangential form, they require rotation of the momentum residuals into this form. This was a new feature implemented in ARIA for this project this year.

The geometry is vented in the upper right had corner, where we have also added an outflow boundary. At this outflow boundary, we apply no boundary condition allowing the natural boundary condition to guide the flow out of the mold.

3.4.2. Initial Conditions

The level set requires an initial condition defining the level set zero and the fluid-gas interface, which indicates the amount of foam precursor material that is injected into the mold before

foaming begins. This initial condition is shown in Figure 17 and was chosen to roughly match the experimental initial condition.

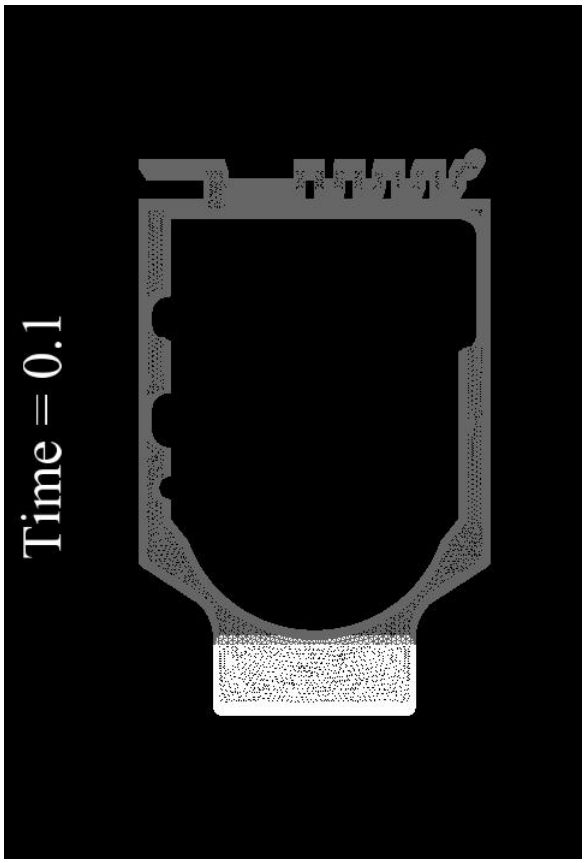

Figure 17. Initial Condition for Level Set Function

The temperature of the foam precursor is mixed to 43^{o}C. However, the injection port is located very close to the aluminum mold wall and the material heats up rapidly to a temperature of 52.7^{o}C. Thus, we could use either temperature for our initial condition.

The initial condition is estimated from mixing and injection times to be roughly 0.6.

3.5. Mass Conservation

Level set methods are not inherently mass conserving. For the finite element method in general, mass conservation occurs on a global basis, not a local one, since our shape functions are C_o and continuous across element boundaries. This can lead to areas with mass sources and sinks, which then lead to either increase or decrease in the mass of phase A and B. Foam self-expansion can be particularly susceptible to mass loss issues compared to incompressible flow problems since volume and velocity generation occur via the continuity equation, which also includes the stabilization terms. Mass conservation can be estimated in the high-end visualization tool

Ensight [Ensight, 2008] by integrating the density over the foam phase to get the mass as a function of time and comparing this to the and initial mass.

Some sources of mass loss and amelioration schemes are listed below:

1. Surface tension – use least square projection for normal reconstruction.
2. Time integration – use constant, small time step (Δt = 0.05s-0.10s for foam rise on the order of 200s-500s). Variable time stepping seems to aggrevate mass loss issues, probably related to level set advection equation.
3. Pressure stabilization – use Dohrmann-Bochev stabilization on $\frac{\partial p}{\partial t}\Delta t$ instead of p alone as this seems to handle pressure jumps better than two discontinuous pressures, which seem to be underspecified for a bilinear velocity interpolant.
4. Level set advection scheme – use Taylor-Galerkin to improve performance of advection operator as velocity is increase. Make sure you have your time step size set that you are below the Courant limit, $\frac{v\Delta t}{h_{elem}} \leq 1$, where Δt is the time step size, and h is the element size [Finlayson, 1992]. In practice, we actually set the Courant number to be less than 0.25.
5. Renormalization – use constrained renormalization, and limit the number of times your renormalize to greater than every 3 time steps but no less than 20.
6. Leaky boundary conditions – use rotated boundary conditions with edge conditions (these are under development).
7. Diffuse interface interacting with small geometric features and large curvatures – adaptive mesh refinement may help with this.

The most important components for successful mass conservation of the foam phase are determined by observation to be 1) a constant time step for the time integration and 2) non-leaky rotated boundary conditions with slip 3) transient implementation of PSPP. The current implementation of the rotated boundary conditions demonstrates some leakage at sharp corners, an issue we are working on for future releases of ARIA.

For the time-dependent density function, using renormalization every 3 times step maximum, based on checking the gradient of the level set function, mass is conserved to 2%. For the full model with all variable properties and a large, increasing shear viscosity, the mass conservation is worse at about 10%. We are still investigating why these differences exist and what can be done to improve mass conservation for variable properties and large viscosity jumps between phases.

4. RESULTS

In this section, we summarize the experimental validation studies in the QA fixture and the results from the finite element/level set modeling within ARIA.

4.1. Experimental Validation Studies in QA Fixture

A series of frames from a video taken of a test with the oven at a nominal temperature of 65°C is shown in Figure 18.

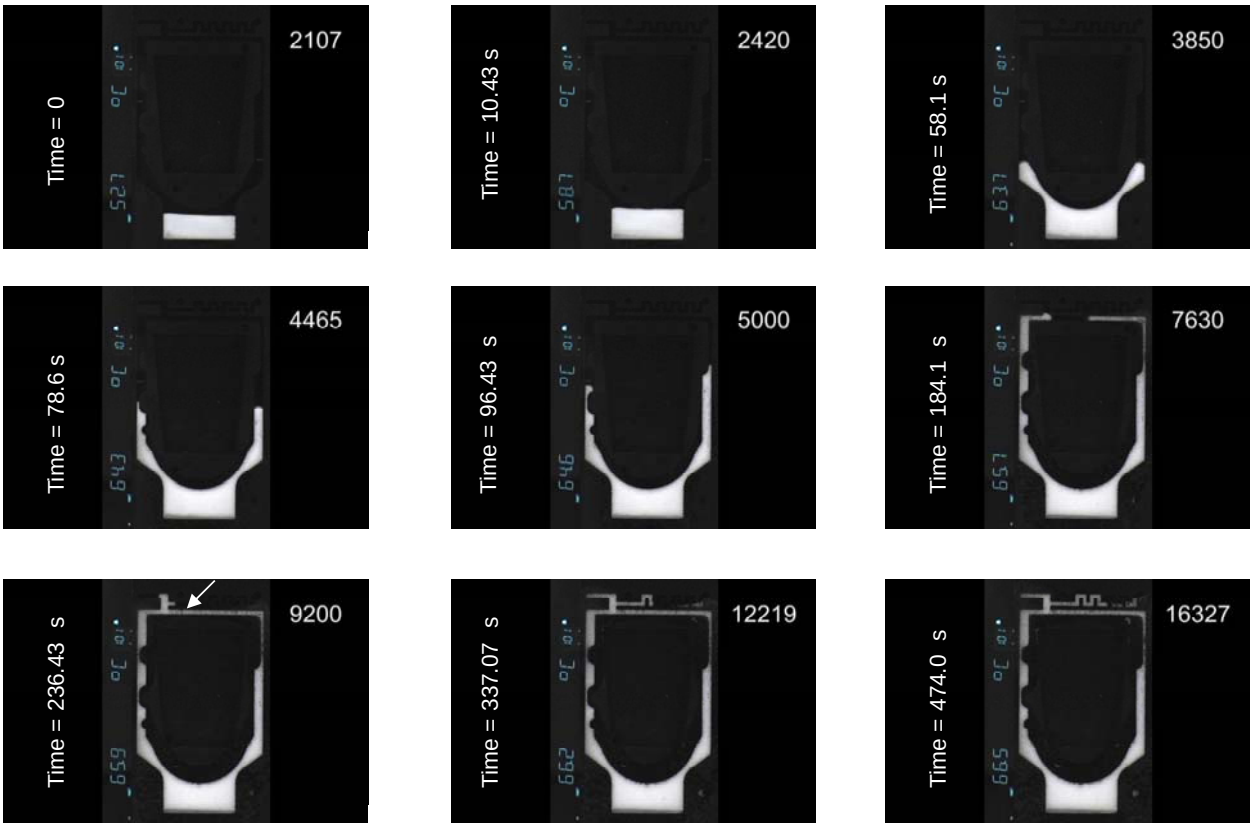

Figure 18. Experimental validation data for QA test fixture

These can be used to validate the model's prediction of the shape and location of the leading fluid front with time. The frame stamp is in right top corner. The frame rate is 30 frame/s. The temperature shown in the video (lower left, sideways) is being measured by thermocouple TC101 in foam near the injection port near the bottom left of the image. Here, we take time zero, when the computational model starts, to be at the end of the injection procedure. Foam hits the bottom of the inner curve at 10.43 s. At 58.10 s, both arms of the front hit the first corner on the inner wall, approximately the height of the thermocouple TC103 on the right side. By about 100 s the foam in the left channel has reached the level of thermocouple TC102 and TC104. The right channel fills faster than the left, and at 117.43 s the right arm has started around the top corner. At 236.43 s the two arms just touch, and an arrow points just to the right of the knit line. By 337.07 s, foam has appeared in the right corner, seemingly spontaneously. This seems to be an experimental artifact, possibly caused by a leak in the front cover, although it may be that a large

bubble has appeared between the upper right corner and the square wave section of the channel. Also of interest is that a void appears at the dead-end on the upper left hand side of the mold, but disappears quickly either by leaking out of the mold or by diffusing into the foam.

Temperature monitoring from the four thermocouples mentioned in the previous paragraph are shown in Figure 19.

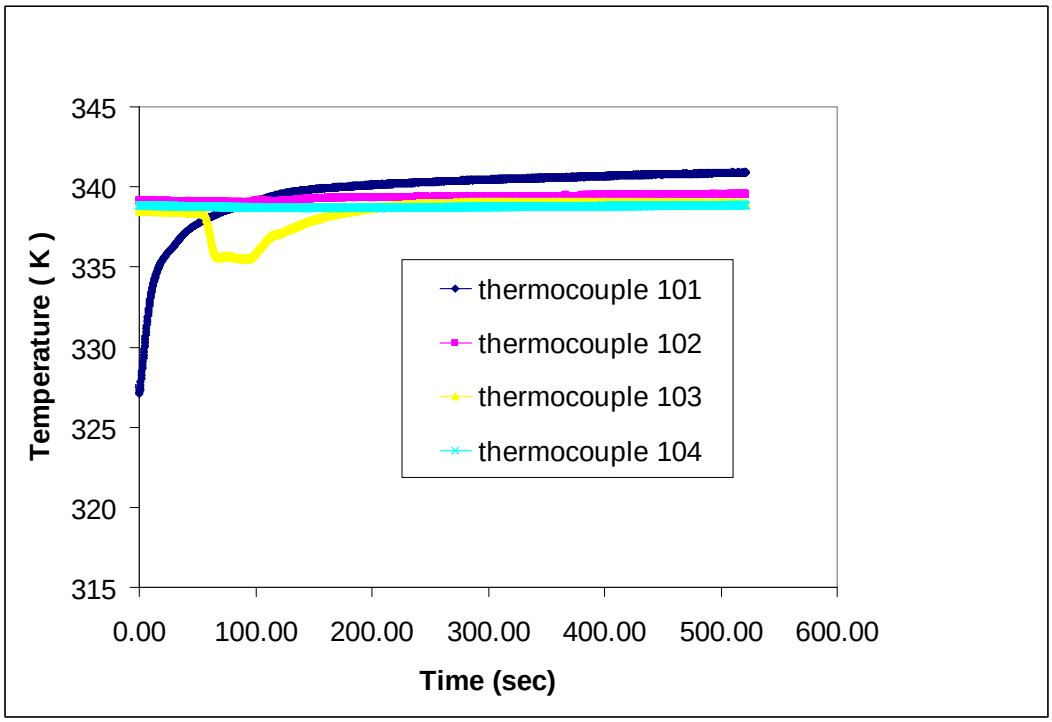

Figure 19. Temperature as a function of time at various locations in the QA test fixture.

From this figure, we see that the temperature at the thermocouples high up in the mold, *e.g.* TC104 and TC102 stay at a constant temperature of 65°C (338.15K), the oven temperature, throughout the experiment. TC101 is located at the inflow and so records the lowest temperature. Because the inflow is so close to the preheated aluminum mold, the material heats up from its mixing temperature of 43°C (316.15K) to 52.7°C (325.85K), which is the temperature we see at the end of injection, when the modeling begins. Away from the walls, the material does not heat up as quickly as seen from thermocouple TC103, which is at the oven preheat temperature of 65°C before the flow front reaches it. Once the cooler foam hits TC103, the temperature drops and then begins to heat up again from the hot oven and mold.

Another important component for the model is the final density in the mold. Here we measure the final density through post-test x-ray CT, the results of which are shown in Figure 20 [Thompson, 2008].

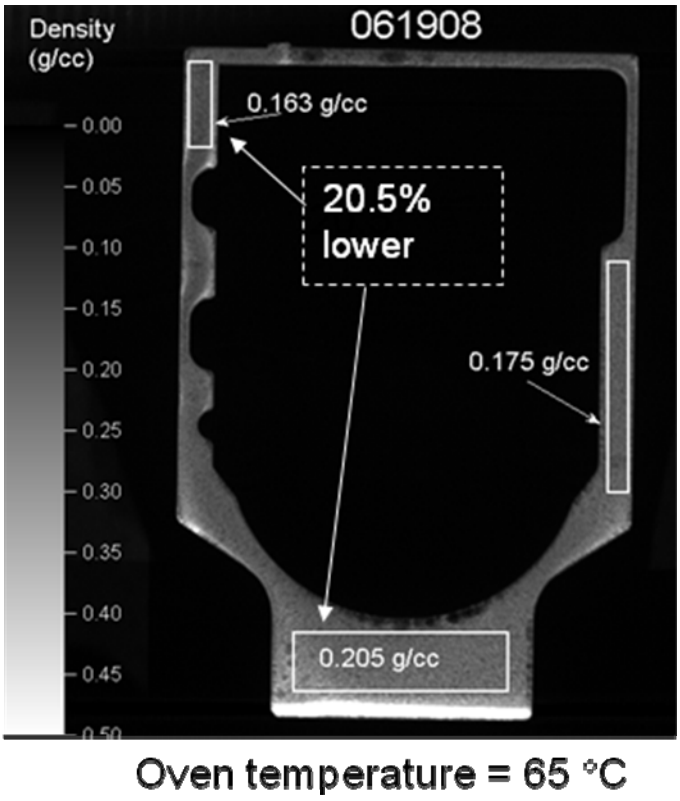

Figure 20. X-ray CT of a part giving density of the final part. The value in the box outlined by a dotted line represents the difference between the lowest and highest boxed region density values.

From the CT data, we can see that the final density varies from 0.163 g/cc at the top left hand corner of the mold to 0.205 g/cc in the bottom of the mold. A layer of unfoamed material has also settled to the bottom of the mold. This data shows at least a 20% variation in foam density within the mold. An average value of foam density of 0.185 g/cc was used for the modeling work.

4.2. Finite Element Results

Here we examine the modeling results for three different constitutive equation formulations. The first is a simplified model including the time-dependent density model discussed in previous sections, a constant viscosity, and an isothermal domain. The second includes temperature effects and uses all the complex, material models discussed in Section 2 for viscosity and thermal properties using the time- and temperature-dependent density model. The third uses the full model except simplifies the viscosity model to include only the curing epoxy and not the effects of gas bubbles.

4.2.1. Isothermal, Time-Dependent Density Model

We can compare the results of our simplest foam self-expansion model using a constant viscosity, and ignoring curing, heat transfer, and the dilatational viscosity. We can look at the

shape of the interface marked by the level set zero and compare it with the results from the flow visualization studies (Figure 21).

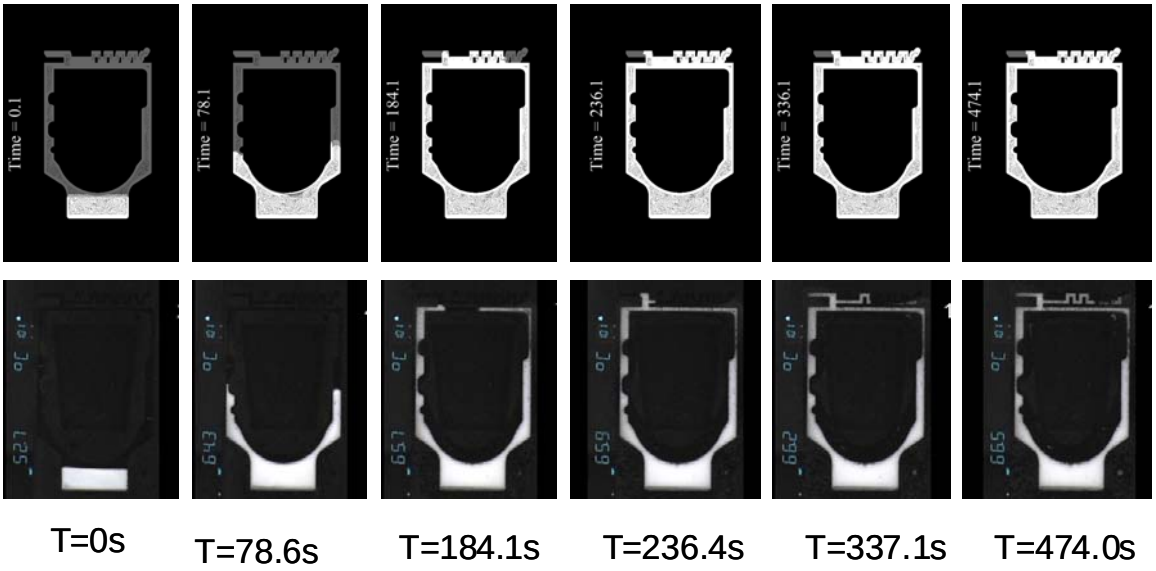

Figure 21. Comparison of ARIA results with flow visualization data. Simulations fill slower initially than the experiments, than fill faster at later times.

From this figure we see the simulations show fairly good agreement with the experiments, with a few qualifications. First, the rate of filling for the simulations seems slower than the experiments. Second, once the material reaches the top of the mold, the simulations become faster than the experiments. Thus, the rate of filling is slightly different in the experiment than in the model.

We also compare the time and location for the two arms of the flow front to come together and form a knit line. This is seen in Figure 22.

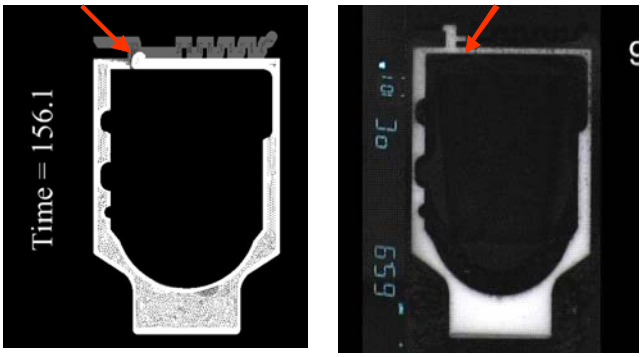

Figure 22. Time and place where knit lines come together for ARIA simulations for time-dependent density model and experimental flow visualization: Left is simulation at time=156.1s and right is data at time=236.4s. Red arrows indicate the place where knit lines join in the mold.

From this figure, we can see that the experiment has its knit line form at 236.4 s while the simulation does this at 156.1 s. In addition, the location of the front coming together is different in the simulation, where the right arm is much faster than the experiments. From viewing the movie and the stills, we find that there are several discrepancies between experiment and model. First, the upper right hand channel looks much narrower in the experiment, and because of the fineness of the channel we find single bubbles moving though this part of the domain like red blood cells in a capillary. In other words, the bubbles are displaying non-continuum effects not accounted for in our model where we need to have at least 6-10 bubbles across the domain to assume a continuum. Another issue that can affect the ARIA results is the choice of slip model and the value of the Navier-slip coefficient. Using a Navier-slip coefficient from the PIV data lead to an entrained gas layer at the solid surfaces. However, we may be getting too much slip at the boundary, and we may need to investigate other wetting models. Other discrepancies between model and experiment occur in the density model itself and the choice of parameters to populate it, which will be discussed in the following section.

We compare the volume in the mold as a function of time for the simulation and experiment in Figure 23.

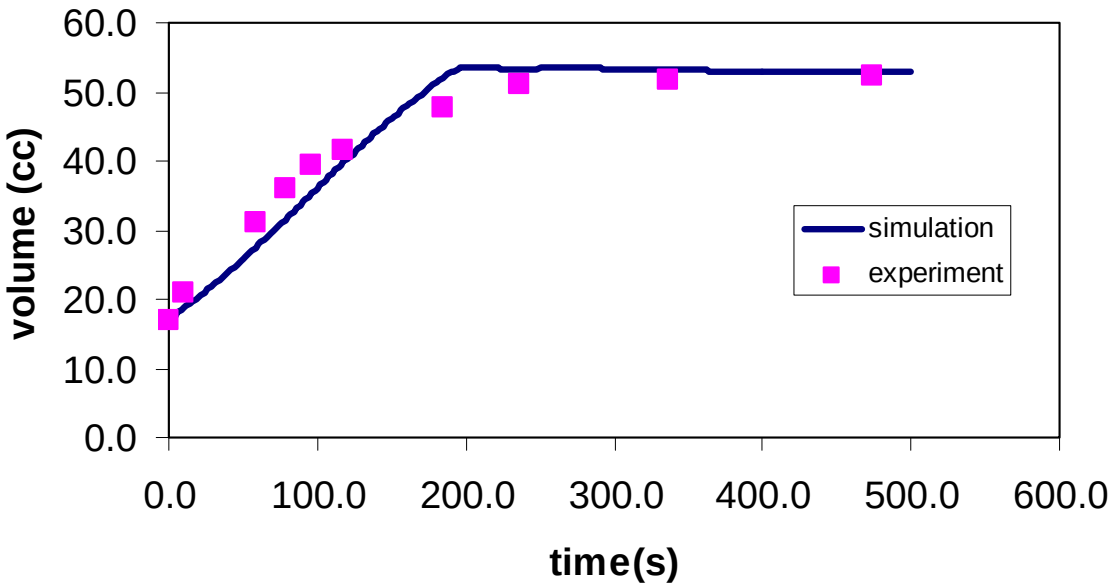

Figure 23. Comparison of volume as a function of time for the ARIA results and flow visualization experimental data [Grillet, 2008]. Experiments fill faster than simulation at early times and then fill slower than simulation at later times.

Here again, we can see that the simulation is slower than the experiment at early times and then reverses itself by being faster than the experiment at late times. The sinusoidal pattern at the top of the mold fills almost instantaneously for the simulation, which may be a numerical issue related to the frequency of level set renormalization, an artifact we are investigating. The model predicts a full mold at about 200 s, while the experiment is as full as it is going to get between 400 s and 450 s. Very little change in volume is observed between 240 s and 474 s, so the model may not be too far off. In the end, the overall match between volumes for the model and data is

extremely good for an engineering model and should be sufficiently accurate for design studies to determine location of voids, knit lines, and fill times, and to optimize vent and gate locations.

4.2.2. Nonisothermal, Time- and Temperature-Dependent Density Model, Full Variable Property Models

We have also tested the full model with variable shear and bulk viscosity, polymerization, heat transfer, and variable thermal properties as discussed in the continuum model section. The results are shown in Figure 24.

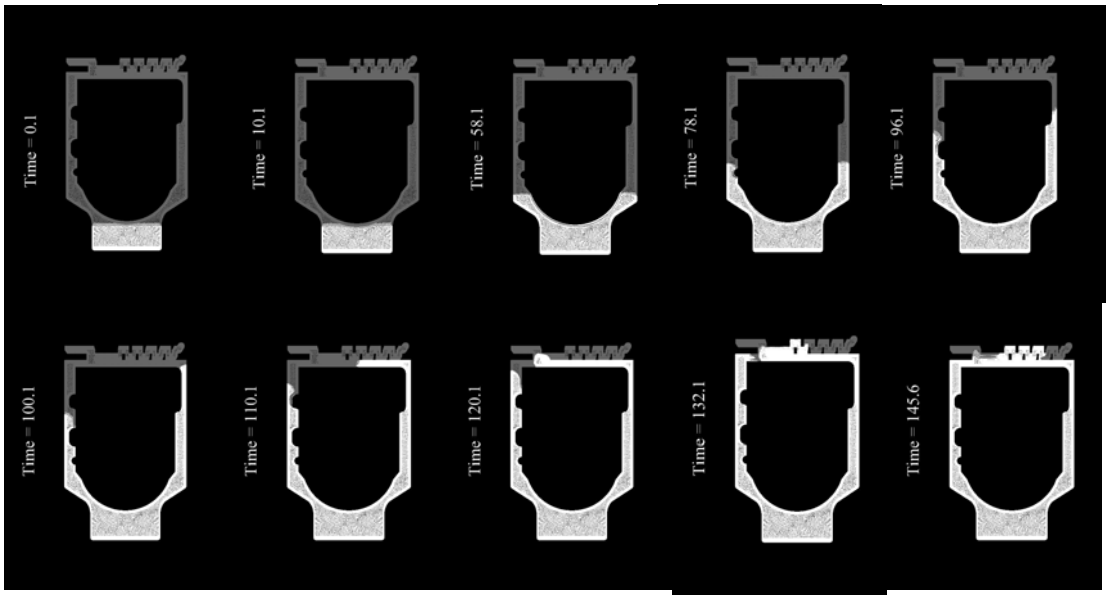

Figure 24. Simulation results for foam self-expansion as a function of time using full model with all variable properties.

For the more complete model, we also see an overall faster fill time than the experiments. This simulation could not be taken to 500s as was done in the previous section. Numerical issues arose as the viscosity increased three orders of magnitude, leading to the loss of convergence of the method. The level set method is well-known for experiencing numerical problems if the ratio of the fluid viscosity to the gas-phase viscosity increases above 1000 [Rao, 2006]. Future methods seek to address this issue by implementing a sharp-interface method in conjunction with enriched nodal variables along the interface.

Because this simulation did not run as long as the experiment, we have a limited number of time planes to compare (Figure 25).

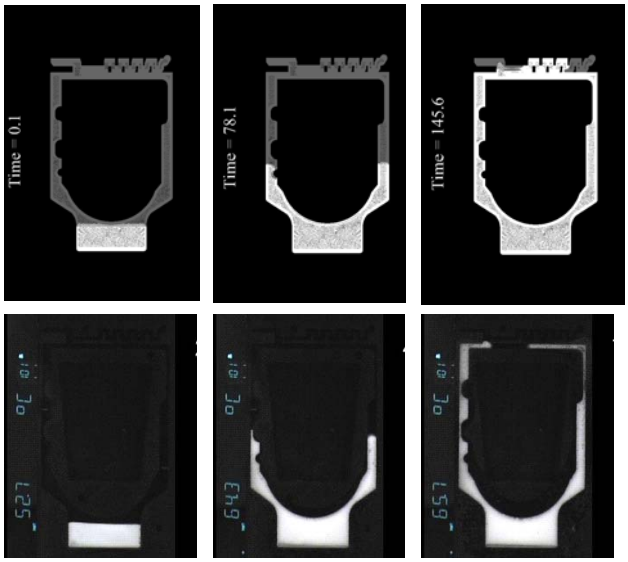

Figure 25. Comparison between simulation (top) using the full model with variable properties, heat transfer, and polymerization, and experiment (bottom).

This figure clearly demonstrates that the full model is filling much faster than the experiments, similar to the previous model but with an estimated fill time of 160 s: considerably more quickly than the previous model's fill time of 200 s.

We compare the time for knit lines to come together as seen in Figure 26.

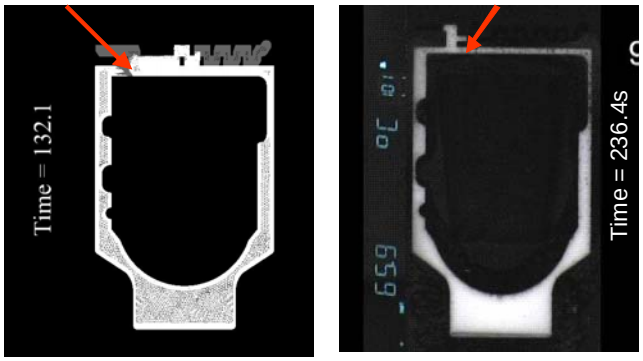

Figure 26. Time and place where knit lines come together for ARIA simulations for full model and experimental flow visualization: Left is simulation at time, 132.1 s, and right is data at 236.4 s. Red arrows indicate the place where knit lines join in the mold.

Again, the knit lines are coming together much sooner for the simulation than the experiment and also in different places. The reasons for this discussed in the previous section are also pertinent here.

We can examine the volume as a function of time from simulation and experiment (Figure 27). Here we have also added the theoretical volume calculated from the density model used in the

simulations, using an initial density of 1.14 g/cc and a final density of 0.2 g/cc and an exponent of 1/70 s at 65°C.

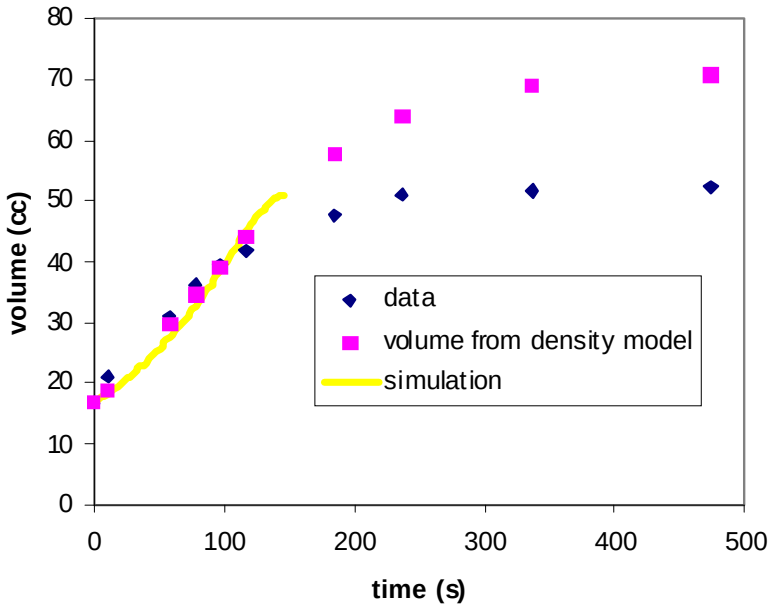

Figure 27. Comparison of volume as a function of time for the full ARIA model and flow visualization experimental data. Experiments fill faster than simulation at early times and then are slower than simulation at later times, though numerical issues ended the simulation early. The theoretical volume from the density model used in simulations is also plotted for reference (pink squares).

By comparing the theoretical volume from the density model used in the simulation, we see that this model is inconsistent with experimental volume data. This implies that the final volume determined from x-ray CT is inconsistent with the volume versus time data. We can determine a density function consistent with this data by varying the parameters until they match the data, as seen in Figure 28. To match the volume data the initial density is 1.14 g/cc, the final density is 0.36 g/cc and the time dependent exponent is 1/55 s at 65°C. This final density is consistent with the density determined from converting the initial mass to the volume obtained from Figure 27.

We can hypothesize that the inconsistency between the x-ray CT density and that determined from other analyses arises from the fact that the CT data is calibrated for densities of foam in the range of 0.1-0.3 g/cm^3. However, the bright zone in the bottom is pure unfoamed liquid, possibly containing a higher concentration of fluorinert than the original mixture since fluorinert is denser than epoxy resin and has a tendency to settle. This bright zone in the CT could have a density of 1.1-1.4 g/cm^3, which would be far outside the calibration range.

There is obviously uncertainty in the initial and final density data. The initial density can range from 0.9-1.14 g/cm^3. The final can range from 0.2-0.3 g/cm^3, but can also vary spatially from 0.15-1.4 g/cm^3. The difference between the initial and final density determines the rate of filling

for the simulation. The larger the difference, the faster the mold will fill. The higher the initial density, the more material there is available for filling the mold.

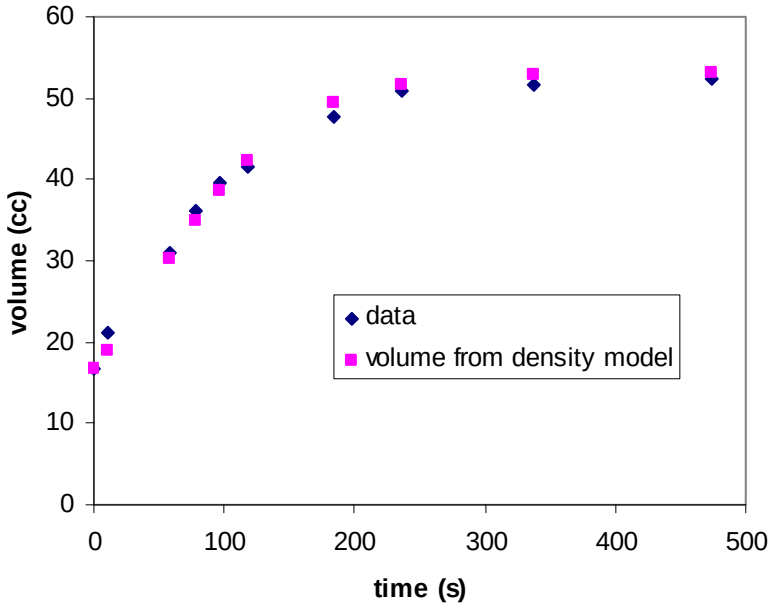

Figure 28. Volume as a function of time for flow visualization experiments (blue) fit to a theoretical volume, based on a consistent density model (pink squares). To match the volume data the initial density is 1.14 g/cc, the final density is 0.36 g/cc and the time dependent exponent is 1/55 s at 65°C.

This density model should fill the mold more slowly and match with experiment better if we used it; however, we strive to populate our material models from independent experiments and not the validation data. The issue of populating the density model seems to be rife with uncertainty and needs to be investigated further.

From the full model, we can examine the property variations in the foam as a function of time for density, shear viscosity, bulk viscosity, heat capacity, thermal conductivity, and gas volume fraction in Figure 29.

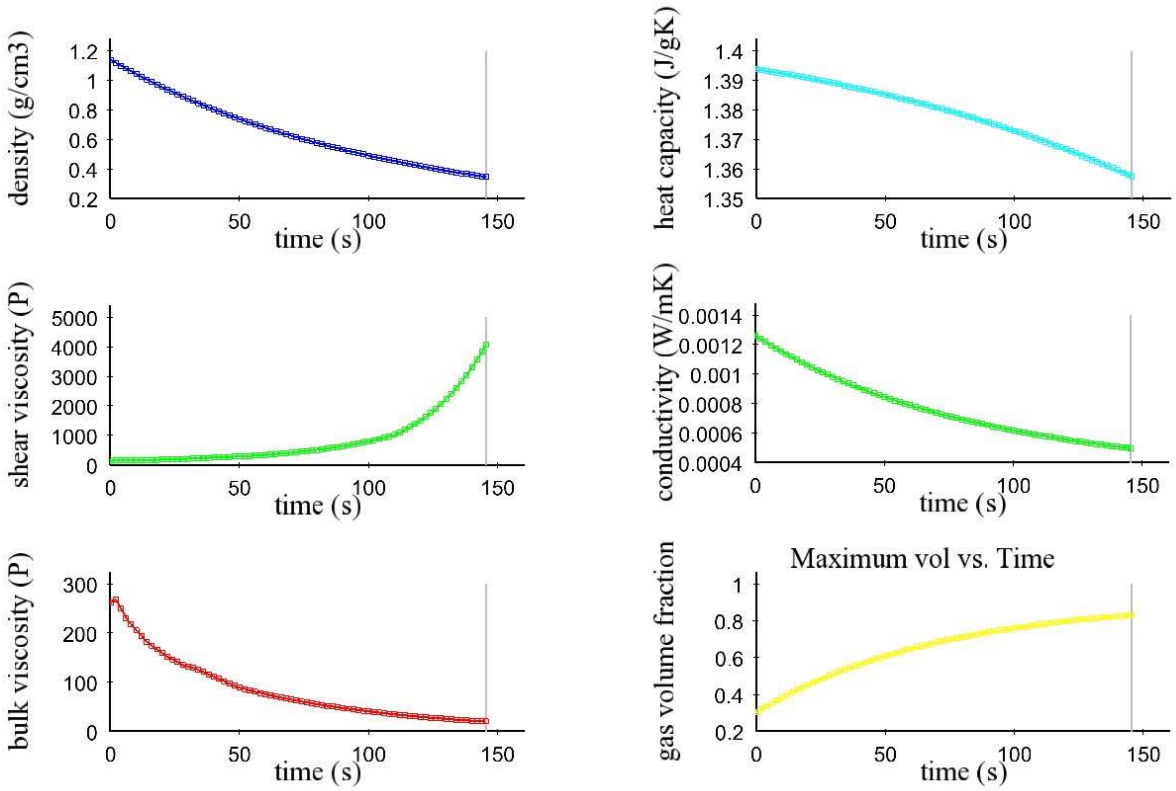

Figure 29. Property variation in the foam as a function of time for density, shear viscosity, bulk viscosity, heat capacity, thermal conductivity, and gas volume fraction.

In Figure 29, the density has decayed to almost its minimum value even though the simulations have only proceeded to 150 s. The shear viscosity begins at a low, uncured value of roughly 40 P and then increases due to curing and bubble evolution resulting in a final viscosity of 4000 P. The effect of bulk viscosity is largest initially, when the gas fraction is small and the divergence of velocity is its largest and decays to a small value at 150 s. The heat capacity is dominated by the continuous epoxy phase and its volume fraction dependence can be safely ignored in favor of a constant value. The thermal conductivity, on the other hand, changes by a factor of three during the blowing process and should therefore be included as a gas volume-fraction dependent function.

Other variables of interest are the extent of reaction and the temperature. The maximum values of these variables are given in Figure 30.

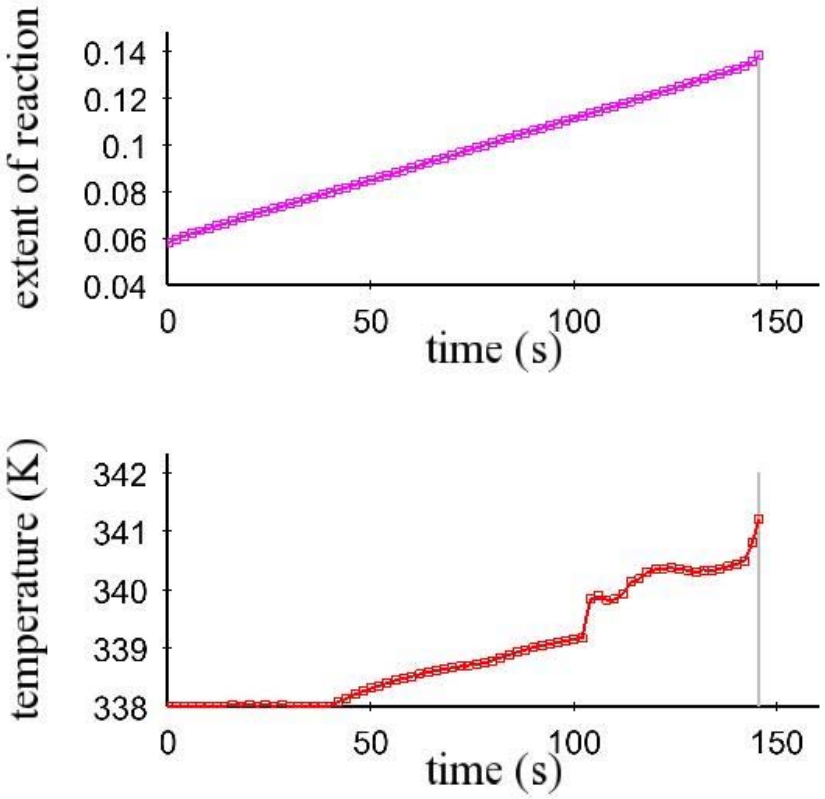

Figure 30. Maximum temperature and extent of reaction as a function of time.

From Figure 30, we can see that the extent of reaction increases fairly linearly from the initial condition of 0.06 and reaches a maximum of 0.14, still far from the gel point of 0.6 but enough to produce heat above the oven temperature of 338 K. We can also look at temperature contours taken on a slice in the center of the mold (Figure 31). Note, if we look at the temperature at the wall it would be quite uninteresting, since it is set to the oven temperature as a Dirichlet condition. From this figure, we can see that that the material starts off at the initial condition of 43^{o}C, which is below the boiling temperature of fluorinert. This implies that the foam precursor in the center bottom section of the mold should not be foaming, which is contrary to our experimental observation that foaming begins immediately. This implies that a more correct initial condition may be that developed from TC101, which is 52.7^{o}C.

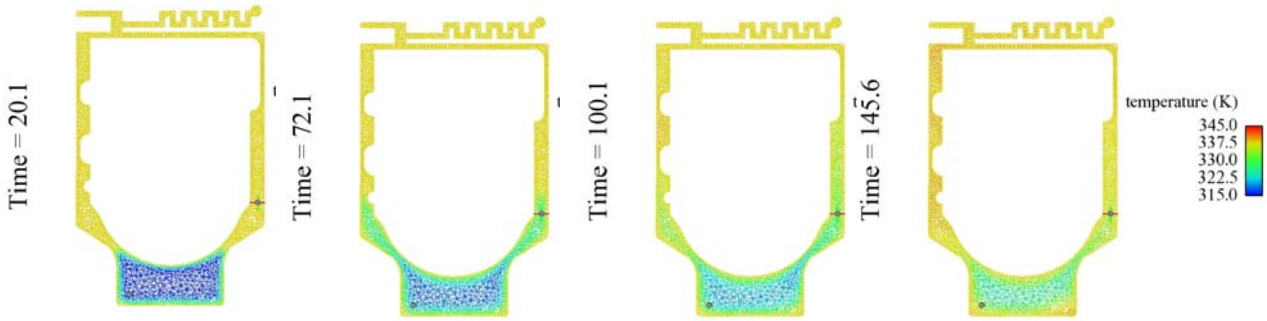

Figure 31. Temperature contours at a slice in the center of the mold.

From Figure 31, we note that the gas phases stays at the oven temperature while the foam material slowly heats up, reaching the oven temperature after 145.6 s.

We also examine the temperature in the ARIA simulations at locations similar to TC101, the inflow, and TC103, the thermocouple located in the side wall. These results are shown in Figure 32 for a simulation using an initial condition for temperature of 43°C. The other two thermocouples are uninteresting as they just track the oven temperature and do not change during the foaming process.

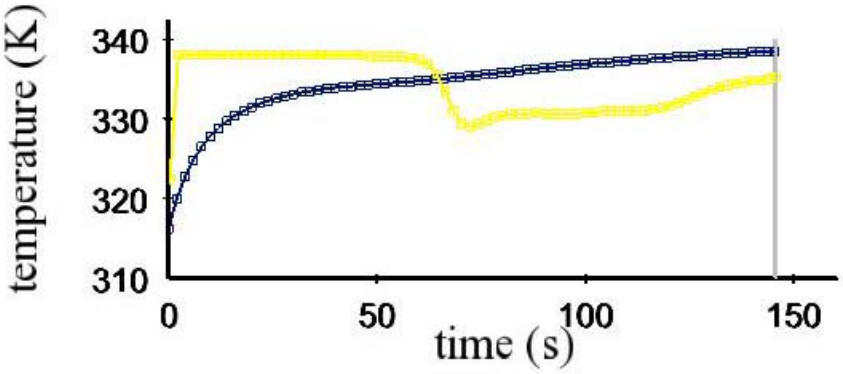

Figure 32. Temperature profiles from simulation for TC101 (blue), TC103 (yellow) for a temperature initial condition of 43°C.

Comparing the results from the simulation, Figure 32, to the experiments, Figure 19, we can see that we do a good job of representing the drop in temperature associated with the cool foam hitting TC103 at roughly 70s and its subsequent heating back up to the oven temperature; though in the simulation TC103 heats back up more slowly than in the experiment. For the inflow thermocouple, TC101, we seem to start off at a much lower temperature, but heat up in a similar manner to the experimental data. This issue could be related to using a wrong initial temperature, *e.g.* 43°C instead of 52.7°C. This thermocouple is very close to the wall and heats up because of

this fact. For areas in the center of the mold, the temperature does not rise as quickly. We will investigate this in the future by adding additional thermocouples to the mold in regions that are potentially colder initially and then hotter due to the exothermic polymerization reaction.

As with the simple time-dependent density model, we feel the full model gives a fairly good comparison to experiment for an engineering model for properties we can measure such as temperature, filling profile, and void location.

We used the full model to investigate an initial condition more consistent with the data. If we use an initial condition of 52.7°C, we get a result that looks more like the experimental data (Figure 33).

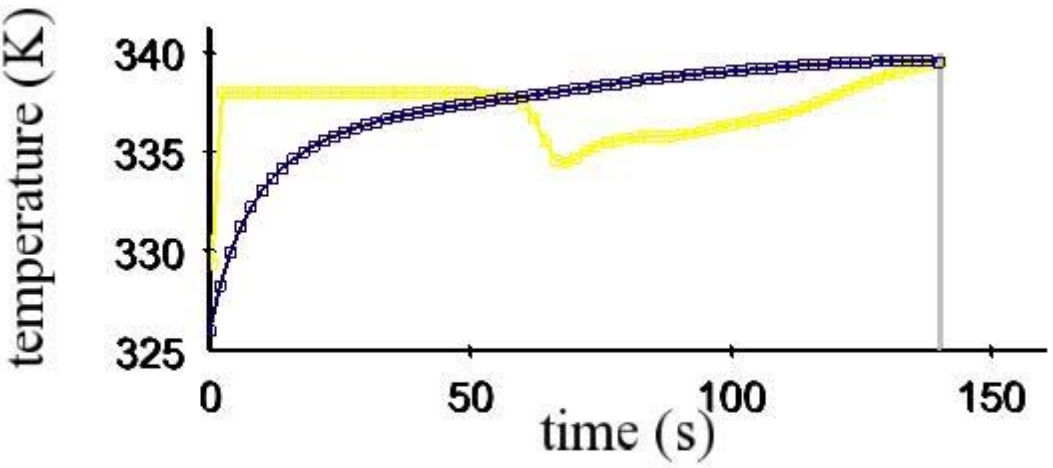

Figure 33. Temperature profiles from simulation for TC101 (blue), TC103 (yellow) for a temperature initial condition of 52.7°C.

4.2.3. Nonisothermal, Time- and Temperature-Dependent Density Model, Variable Property Models with a Simplified Viscosity Model

We have also tested a third model, similar to the one discussed in the previous section but with a slightly different viscosity model. Here, the viscosity varies as the epoxy continuous phase with degree of polymerization and temperature, but we ignore the effects of bubbles on the foam, *e.g.*, equation (11). This seems to be a reasonable model, since it is unclear exactly what viscosity the fluid sees during self-expansion and how different this viscosity is from the one seen in a pressure driven or viscometric flow. Here, the foam expansion leads directly to velocity from the continuity equation and not to a pressure driven flow.

For the simplified viscosity model, the simulation ran as long as the experiment. Comparison between the two is given below for an initial density of 1.14 g/cm^3 and a final density of 0.33 g/cm^3 (Figure 34).

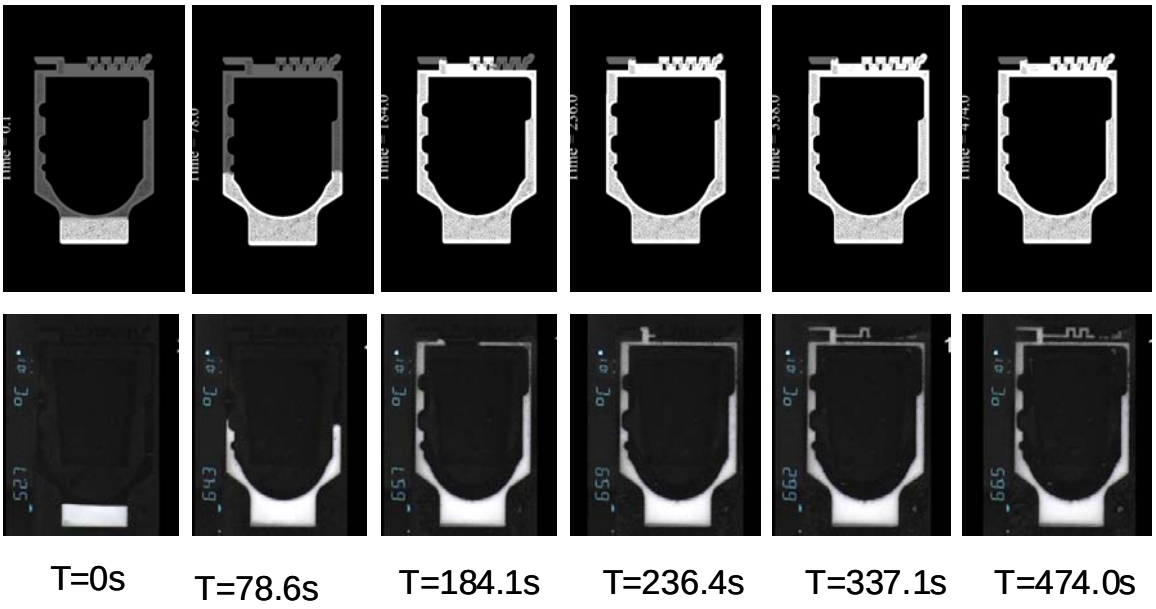

Figure 34. Comparison between simulation and experiment for full model with variable properties, heat transfer and polymerization, but using a curing epoxy viscosity model without the effects of gas bubbles.

This figure clearly demonstrates that the full model with the simplified viscosity is filling much faster than the experiments, similar to the previous model but with an estimated fill time of 200s: similar to the time-dependent density model. Because the difference between the initial and final density is a smaller than the previous simulation, the mold fills more slowly even with the same exponent.

We can also compare the time for knit lines to come together as seen in Figure 35.

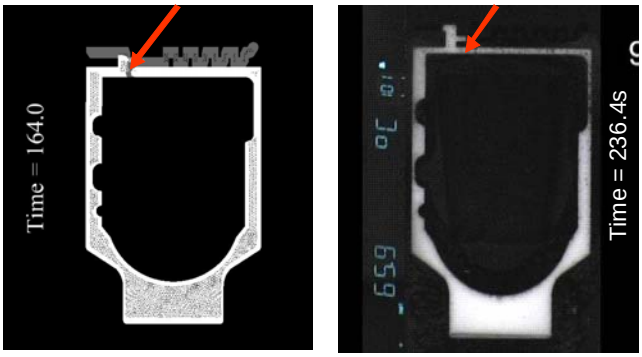

Figure 35. Time and place where knit lines come together for ARIA simulations for full model and experimental flow visualization: Left is simulation at 164.0 s and right is data at time, 236.4 s. Red arrows indicate the place where knit lines join in the mold.

Again, the knit lines are coming together much sooner for the simulation than the experiment, but in a similar place for this slower and lower viscosity simulation. The reasons for this mismatch are discussed in the previous sections are also pertinent here.

We can examine the volume as a function of time from simulation and experiment (Figure 36).

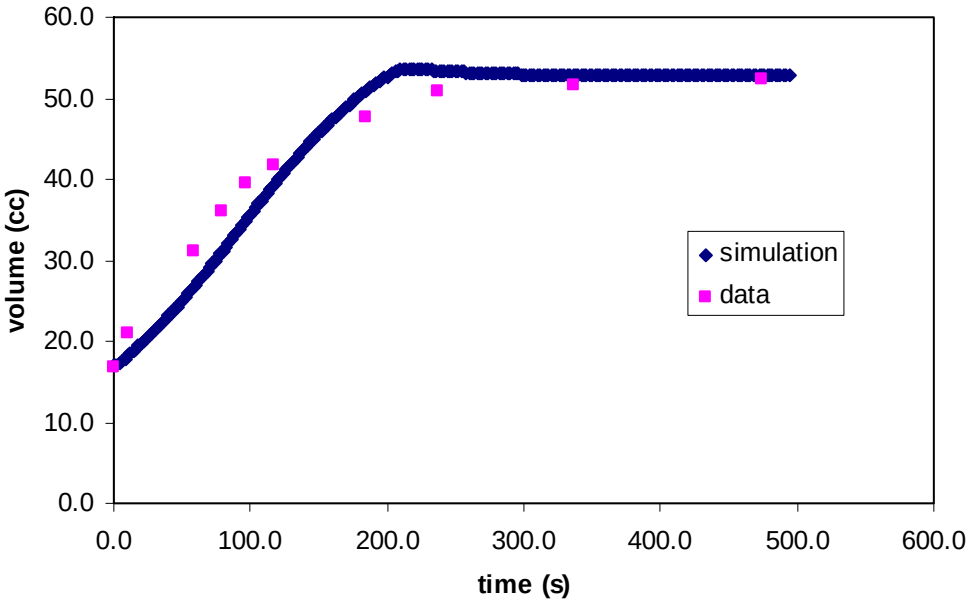

Figure 36. Comparison of volume as a function of time for ARIA simulations and flow visualization experimental data. Similar to the other models, the experiments fill faster than the simulation at early times and then slower than the simulation at later times.

This plot looks very similar to the time-dependent density model with the same caveats of the simulation filling too slowly initially and then too fast at the end.

We can also examine the material properties for the reduced viscosity model (Figure 37). The density, heat capacity, conductivity, and gas volume fraction look very similar to the previous simulation results (Figure 29). The shear viscosity is much smaller and has a maximum of 200 P instead of 4000 P for the simulation where the effect of gas bubbles was included. Because the shear viscosity is so much lower than in the previous simulation, we can run the simulation to 500 s without incurring the numerical difficulties associated with the higher viscosity case.

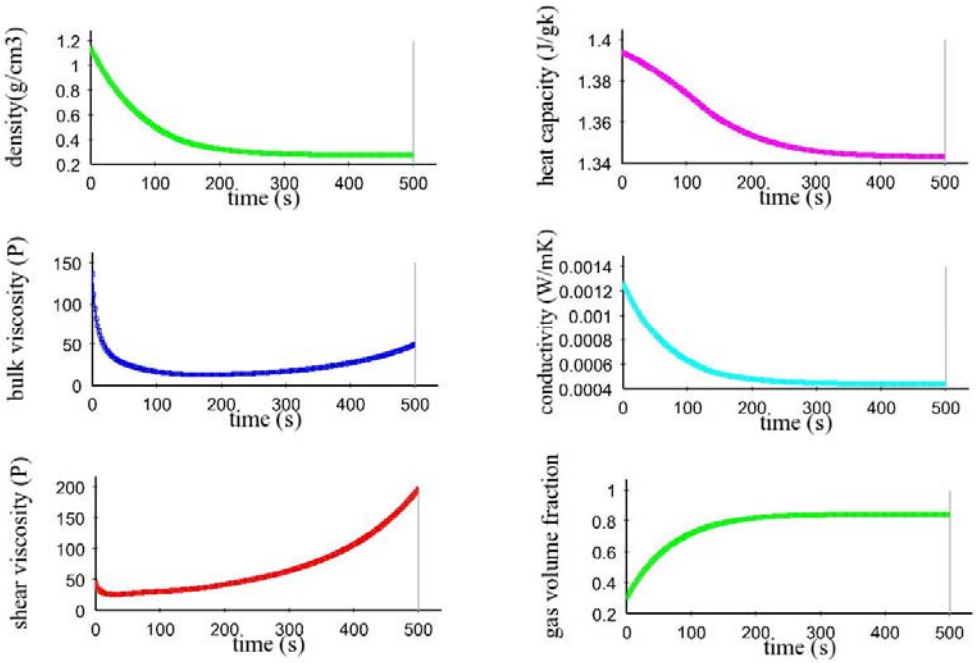

Figure 37. Property variation in the foam as a function of time for density, bulk viscosity, shear viscosity, heat capacity, conductivity, gas volume fraction for simpler curing viscosity model.

Since the thermal properties are unchanged, the thermal results should be similar. To determine the effect of the exotherm, in future, we would have to carry the simulation out past the filling stage. We have done similar work before on the neutron generator where the full fluid-thermal-curing simulations were carried out until the epoxy reached the gel point, at which point a thermal-curing simulation was begun to determine the maximum temperatures attained in the mold. It would be good to also have our validations experiments extended through cure.

As discussed before, the model could be improved by reexamining the curve fit for the density function, as this is at the heart of the model.

5. CONCLUSIONS AND FUTURE WORK

We have developed a production scale model, which can be used to study foam self-expansion in complex geometries. The model is based on a finite element/level set implementation in the ARIA module of the SIERRA mechanics framework. ARIA can be run either in serial or massively parallel modes. The foam model is based on continuum theory that homogenizes the effects of gas bubbles into a time-dependent density model. ARIA was run on 68 processors of Thunderbird to obtain results for a simpler foam self-expansion model in the QA test fixture, where the density is variable, obeying an exponential decay function, the viscosity is Newtonian, dilatational effects are ignored, and the domain is assumed to be isothermal with including polymerization of the epoxy. A second model was run in the QA test fixture that included nonisothermal effects, in addition to curing, variable properties, and dilatational viscosity. The results from the models were compared to flow visualization data. Both models performed fairly well, though there were some quantitative differences between the filling rates and shapes thought to come from uncertainties in the density model.

This is an engineering design tool that will allow us to study processing variable such as temperature and fluorinert and air concentrations, as well as the effect of gate and vent location for optimizing the foam encapsulation process. It has many good features: 1) It is reasonable robust numerically, especially the time-dependent density model 2) It is fast enough that it can be run in complex geometries of real components such as that of the firing set and AFS. 3) It is fairly easy to understand and populate via simple experiments and experimental analysis, and can be applied to other foams such as REF and even polyurethanes, 4) It can predict trends for engineering optimization.

However, there is obviously room for improvement and we would like to develop a more quantitative model. We are currently developing a detailed treatment of the fluorinert nucleation and growth that does not rely wholly on the density model for the correct foaming. Missing from our current model is this effect, along with the density gradients created by the buoyancy of the gas bubbles relative to the epoxy that result in creaming. For future work, we plan to incorporate a fluorinert nucleation and growth model based on cavitation theory and cloud-physics theory, where nucleation is thought to occur from collisions of gas bubbles rising with fluorinert droplets settling (see Figure 34 for an illustration).

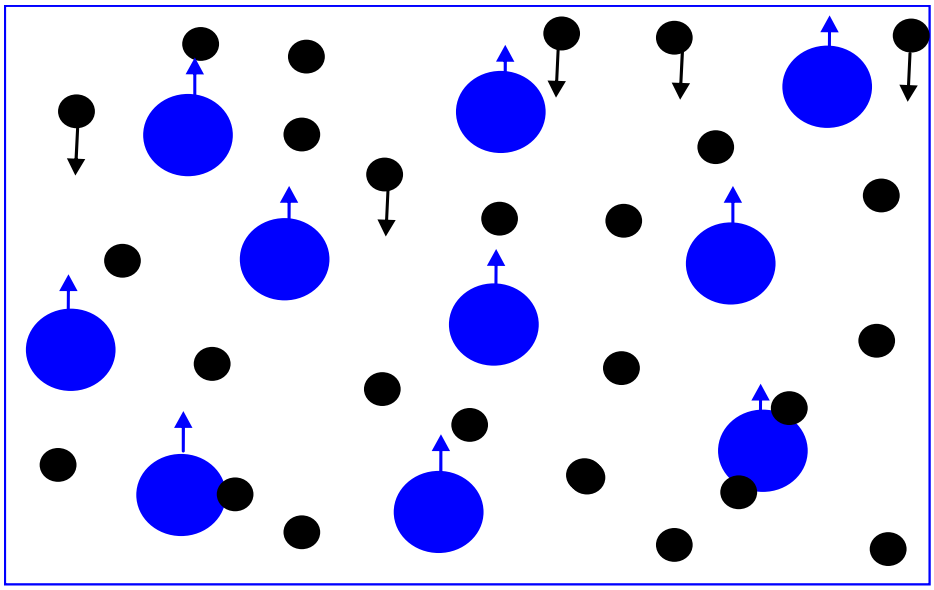

Figure 38. Conceptual representation of the foam nucleation process. Blue bubbles represent the gas phase. The black bubbles represent fluorinert liquid-phases inside a continuous epoxy phase, shown as white. Due to the different densities between the black and blue phases, there exists a relative motion leading to enhanced collisions.

We will include the effects of bubble transport using a model similar to models for glass microballoons in the neutron generator, which proved useful for predicting the complex flow of floating microballoons in curing epoxy in complex geometries [Rao et al, 2007].

6. REFERENCES

D. B. Adolf, *Measurement Techniques for Evaluating Encapsulant Thermophysical Properties During Cure*, SAND96-1458, Sandia National Laboratories, Albuquerque, NM June 1996.

D. B. Adolf, J. E. Martin, R. S. Chambers, S. N. Burchett, T. R. Guess *J. Mater*. Res, 1998, *13,* 530.

G. K. Batchelor, *An Introduction to Fluid Mechanics*, Cambridge University Press, New York, 1967.

R. B. Bird, W.E. Stewart, E.N. Lightfoot, *Transport Phenomena*, John Wiley and Sons, New York, USA, 1960.

J. U. Brackbill, D. B. Kothe, and C. Zemach, *J. Comp. Phys.*, 100, 335 (1992).

R. Clift, J. R. Grace, and M. E. Weber, *Bubbles, Drops, and Particles*, Academic Press, New York, New York, USA, 1978.

J. Donea, "A Taylor-Galerkin Method for Convective Transport Problems," *Int. J. for Num. Methods in Eng.,* 20, 101-119 (1984).

C. R. Dohrmann and P. B. Bochev, "A stabilized finite element method for the Stokes problem based on polynomial pressure projections," *Int. J. Num. Meth. Fluids*, 46:183–201, 2004.

H. C. Edwards, *SIERRA Framework Version 3: Core Theory and Design*, SAND2002-3616, Sandia National Laboratories, Albuquerque, NM, November 2002.

Ensight, Computational Engineering International, www.ensight.com, 2008.

J. D. Ferry, *Viscoelastic Properties of Polymers*, p. 290, Wiley, NY, USA, 1980.

B. A. Finlayson, *Numerical Methods for Problems with Moving Fronts*, Ravenna Park Publishing, Seattle, USA, 1992.

Fluorinert Electronic Liquids: For Electronic Reliability Testing, 3M Product Data Sheet 98-0212-2158-9, Specialty Fluids, 3M Specialty Materials, 3M Center, Building 223-6S-04 St. Paul, MN (1999).

A. M. Grillet, Sandia National Laboratories, Personal Communication, Sept. 2008.

L. J. Gibson, M. F. Ashby, *Cellular Solids: Structure and Properties*, 2nd Edition, Cambridge Solid State Science Series, Cambridge University Press, Cambridge, UK.

N. C. Hilyard, A. Cunningham, *Low Density Cellular Plastics: Physical Basis of Behavior*, Chapman & Hall, New York, USA, 1991.

T. J. R. Hughes. *The Finite Element Method*. Dover Publications, New York, USA, 2000.

D., Jacqmin, (1995) "Three Dimensional Computations of Droplet Collisions, Coalescence, and Droplet/Wall Interactions Using a Continuum Surface-Tension Method," AIAA-95-0883.

A. M. Kraynik, T. A. Baer, M. Martinez, Sandia National Laboratories, Personal Communication, Aug. 2005.

J. Mahoney, Honeywell Federal Manufacturing and Technologies, Kansas City Plant, Personal Communication, September 2007.

D. Mao, J.R. Edwards, A. Harvey. *Chem. Engn. Sci.*, 61, 1836. 2006.

J. P. Martin, D. B. Adolf, and J. P. Wilcoxon, *Physical Review A* 39: 1325, 1989.

C. A. May, Ed., *Epoxy Resins Chemistry and Technology*, 2nd Edition, Marcel Dekker, New York, USA, 1988.

L. A. Mondy, R.R. Rao, C. F Brooks, D. R Noble, R. C. Givler, A. C. Sun, A. M. Kraynik, T. A. Baer, E. D. Wilkes, P. K Notz, M. M. Hopkins, A. M Grillet,. R. O. Cote, J. N. Castaneda, D. B. Adolf, J. Mahoney, K. Berchtold, M. Brooks, and A. Graham,*Wetting and Free Surface Flow Modeling for Potting and Encapsulation*, SAND2007-3316, Sandia National Laboratories, Albuquerque, NM, May 2007.

L. A. Mondy, R.R. Rao, H. K. Moffat, A. M. Kraynik, E. D. Russick, D. B. Adolf, A. M. Grillet, C. M. Brotherton, C. Bourdon, A. Gorby, R. Cote, J. Castenada, K. Thompson, J. Mahoney, Experiments for Foam Model Development and Validation, SAND2008-XXXX, Sandia National Laboratories, Albuquerque, NM, September 2008.

P. K. Notz, S. R. Subia, M. M. Hopkins, H. K. Moffat, D. R. Noble, ARIA Manual Aria 1.5: User's Manual, SAND2007-2734, Sandia National Laboratories, Albuquerque, NM, April, 2007.

R. K. Prud'homme and S.A. Khan, p.234, Ch.4, *Foams: Theory, Measurement and Application*, Surfactant Science Series, v.57, eds. R. K. Prud'homme and S.A. Khan, Marcel Dekker, Inc, New York, New York, 1996.

R. R. Rao, L. A. Mondy, D. R Noble, M. M. Hopkins, P. K Notz, T. A. Baer, L. L. Halbleib, p. Yang, G. R. Burns, A. M Grillet,. C. F Brooks, R. O. Cote, J. N. Castaneda, *Modeling Injection Molding of Net-Shape Active Ceramic Components*, SAND2006-6786, Sandia National Laboratories, Albuquerque, NM, November 2006.

H. W. Russell, “Principles of heat flow in porous insulators,” *J. Amer. Ceramic Society*, 18, 1-5, 1935.

D. Seo; J. R. Youn; C.L.I. Tucker *Int. J. Numer. Meth. Fluids, 42*, 1105, 2003.

J. A. Sethian. *Level Set Methods and Fast Marching Methods*, volume 3 of Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, New York, USA, 2nd edition, 1999.

R. R. Rao, L.A. Mondy, S.A. Altobelli, “Instabilities during batch sedimentation in geometries containing obstacles: A numerical and experimental study,” *Int. J. Num. Meth. Fluids*, 55:723–735, 2007.

M. Sussman, and E. Fatemi, “An Efficient, Interface-Preserving Level Set Redistancing Algorithm and its Application to Interfacial Incompressible Fluid Flow,” *SIAM J. Sci. Comput.*, 20(4), 1165-1191, 1999.

K. Thompson, Sandia National Laboratories, Personal Communication, 2008.

Trilinos, http://trilinos.sandia.gov, 2008.

Wikipedia, www.**wikipedia**.org, 2008.

APPENDIX A: FOAM MATERIAL PROPERTIES AND ARIA INPUT FILE

```
Foam properties

${R=1.987} cal/mol K

${ca = 90} degrees
${vw = 1.93e-1}  cm/s
${cg= 1/.497}
${tau= 1.}
${sigma = 39.8 } dyn/cm
${elem_size = 0.01}

${beta = 0.01}
${g = 980.} g/cm3
${m_air = 0. } mass fraction air 2.66E-4 - Harry's uses this
${m_epoxy = 0.954074} mass fraction epoxy

${rho_liq = 1.14} g/cm^3
${rho_gas = 0.001} g/cm^3 should be .00055
${rho_fluor_l = 1.68} g/cm^3
${rho_fluor_g = 1.39E-2} g/cm^3

#Foam density parameters
${rhof_i = 1.14} g/cm^3
${rhof_f = 0.27} g/cm^3
${CT = 116250}
${DT = 274.26}

#Foam viscosity parameters
${mua = 6.e-8} P includes factor of 1.5 from fluorinert
${T_mu = 0. } K In the code this removes the linear term
${mub = 0.0 } P
${nmu = -3.5 }
${mmu = 1. }
${Emu = 1.30e4 } cal/mol
${xi_0 = 0.058 } extent of rxn after injection
${xi_c = 0.6 } extent of rxn at gel pt


${mu_liq = 65 } P #should be ??
${mu_gas = 0.1} P #should be .01

${ergtoJ = 1.e-7} Merg/erg
#Thermal properties
${k_liq = 1.8E4*ergtoJ} erg/(cm sec K) liquid conductivity
${k_air = 2.5E3*ergtoJ} erg/(cm sec K) air conductivity
${k_foam = 4540.*ergtoJ} erg/(cm sec K) foam conductivity Gill and Dobranich,
SAND2002-1769
${cp_foam = 1.48e7*ergtoJ} erg/ g K foam heat capacity Gill and Dobranich,
SAND2002-1769
${cp_liq = 2.0e7*ergtoJ} erg/g K liquid heat capacity
${cp_air=1.02e7*ergtoJ} erg/g K air heat capacity
${h = 2.5e7*ergtoJ} erg/cm^2 K s
${hrxn = -2.23e9*ergtoJ} erg/g Heat of reaction for epoxy cure from Adolf
${hevap = 8.71164e8*ergtoJ} erg/gm Heat of evap for fluorinert
```

```
${T_i = 325.85} K temperature of fluid as it goes in the mold
${T_oven = 338} K for first hour
${T_max = 410} K maximum temperature for density model applicability



#Extent of reaction
${N = 1.4} curing exponent
${E1 = 1.10e4} cal/mol
${a = 8600.} 1/s
```

```
# {include("foam.def")}

begin sierra myJob

   Begin Aria Material multiphase
      Level Set Heaviside = Smooth
      Level Set Width     = Constant width={alpha=0.3*2.54}
      Surface Tension     = Constant sigma={sigma}
      Density             = Phase_Average
      Viscosity           = Phase_Average
      thermal_conductivity = Phase_Average
      specific_heat       = Phase_Average
      Volume_Fraction_Gas = Phase_Average
      Bulk_Viscosity = Phase_Average
   End

   Begin Aria Material fill

      #Density             = Constant rho = {rho_liq}
      #Density              = Exp_Decay Rho_Initial=1.14 Rho_final=0.27
K=0.0125
      Density         =  Foam_Time_Temp  RHO_INITIAL={rhof_i}
RHO_FINAL={rhof_f} C_PARAM={CT} D_PARAM= {DT} T_MAX = {T_max}
      #Momentum Stress     = Incompressible_Newtonian
      #Viscosity       =  Constant  mu = {mu_liq}
      Momentum Stress = Formal_Newtonian
      #Bulk_Viscosity  = Constant  kappa = 0.1
      Viscosity       = Curing_Foam mu_a={mua} T_mu ={T_mu} mu_b={mub}
n_mu={nmu} E_mu={Emu} R={R} ksi_c={xi_c} m_ksi_c={mmu} extent_subindex=0
      Bulk_Viscosity  = Curing_Foam mu_a={mua} T_mu ={T_mu} mu_b={mub}
n_mu={nmu} E_mu={Emu} R={R} ksi_c={xi_c} m_ksi_c={mmu} extent_subindex=0
      Heat Conduction     = Fouriers_Law
      #SPECIFIC HEAT       = CONSTANT  CP = {cp_foam}
      #THERMAL CONDUCTIVITY        = CONSTANT   K = {k_foam}
      SPECIFIC HEAT       = CURING_FOAM CP_G={cp_air} CP_L ={cp_liq} RHO_G =
{rho_gas} RHO_L = {rho_liq}
      THERMAL CONDUCTIVITY        = CURING_FOAM K_G = {k_air}    K_L =
{k_liq}
      species diffusion_0 = Ficks_Law
```

```
      species diffusivity_0 = CONSTANT D = 5.0E-4
      #VOLUME_FRACTION_GAS = FROMDENSITY RHO_LIQUID = {rho_liq}  RHO_GAS =
{rho_gas} RHO_INITIAL = {rhof_i}
      VOLUME_FRACTION_GAS = FromFoamTimeTemp RHO_INITIAL = {rhof_i}
MassFraction_air={m_air} rho_air={rho_gas} rho_vaporFluorinert={rho_fluor_g}
rho_liquidFluorinert={rho_fluor_l}

   End

   Begin Aria Material gas
      Density              = Constant rho = {rho_gas}
      Viscosity            = Constant  mu = {mu_gas}
      Momentum Stress      = Incompressible_Newtonian
      Heat Conduction     = Fouriers_Law
      SPECIFIC HEAT       = CONSTANT  CP = {cp_air}
      THERMAL CONDUCTIVITY        = CONSTANT   K = {k_air}
      species diffusion_0 = Ficks_Law
      species diffusivity_0 = CONSTANT D = 5.e-2
      VOLUME_FRACTION_GAS = CONSTANT value = 0.0
      Bulk_Viscosity       = Constant  kappa = 0.0
   End

   begin trilinos equation solver UMF
      solution method = amesos-umfpack
      Matrix Scaling  = row-sum
      matrix reduction = fei-remove-slaves
   end
   begin trilinos equation solver SUPERLU
      solution method = amesos-superludist
      Matrix Scaling  = row-sum
   end

   begin aztec equation solver iterative
      preconditioning method = dd-ilu
      #solution method = bicgstab
      solution method = gmres
      maximum iterations      = 500
      param-int AZ_kspace value 500
      residual norm tolerance = 1.e-6
      ilu threshold = 1.e-3
      preconditioner subdomain overlap = 2
   end

   begin trilinos equation solver iterative1
      preconditioning method = dd-ilut
      solution method = gmres
      residual norm tolerance = 1.e-6
      maximum iterations      = 500
      param-int AZ_kspace value 500
      ilu threshold = 1.e-6
      ilu Fill = 3
      preconditioner subdomain overlap = 2
   end

  begin aztec equation solver gmres_ilut
      solution        method  = gmres
      preconditioning method  = dd-ilut
```

```
      maximum iterations      = 600
      param-int AZ_kspace value 600
      residual norm tolerance = 1.e-4
      param-real AZ_drop value 1.e-4
      param-int AZ_ilut_fill value 3
      preconditioner subdomain overlap = 2
      Matrix Scaling = row-sum
   end

   begin aztec equation solver gmres_ilu
      solution        method  = gmres
      #solution method = bicgstab
      preconditioning method  = dd-ilu
      maximum iterations      = 600
      param-int AZ_kspace value 600
      residual norm tolerance = 1.e-4
      param-real AZ_drop value 1.e-6
      preconditioner subdomain overlap = 2
      Matrix Scaling = row-sum
   end

   Begin Finite Element Model foam
      database Name = art3d_cm.g
      coordinate system is cartesian
      begin parameters for block block_1
       material multiphase
       phase a = fill
       phase b = gas
      end
   End

   Begin Procedure myProcedure

      Begin Solution Control Description
       Use System Main

       Begin Initialize My_Init
         Advance LS_Region
           Advance Fluid_Region
         #Advance SOLUTION_LEVEL_SET_REDISTANCE
       End

       Begin System Main
          #Use Initialize My_Init
          Simulation Start Time          = 0.0
          Simulation Termination Time          = 500.0
          Simulation Max Global Iterations = 30000
          Begin Transient Stepper
             Transfer Fluid_to_LS
             #Begin Subcycle The_Subcycle_Block
                  Advance LS_Region
              #Event LS_CONSTRAINED_REDISTANCE when "(CURRENT_STEP -
LAST_LS_CONSTRAINED_REDISTANCE_STEP) >= 50 || (LS_GRADIENT_ERROR_NORM(0.) >
0.09 && (CURRENT_STEP - LAST_LS_CONSTRAINED_REDISTANCE_STEP) >= 20)"
              Event LS_CONSTRAINED_REDISTANCE when "(CURRENT_STEP -
LAST_LS_CONSTRAINED_REDISTANCE_STEP) >= 20 || (LS_GRADIENT_ERROR_NORM(0.) >
0.05 && (CURRENT_STEP - LAST_LS_CONSTRAINED_REDISTANCE_STEP) >= 3)"
```

```
            #End
            Transfer LS_to_Fluid
            Transfer PROJ_LS_to_Fluid
            Advance Fluid_Region
            #Advance Fluid_Region when "CURRENT_STEP % 10 == 0"
            #Event LS_CONSTRAINED_REDISTANCE
            Event LS_COMPUTE_SIZES
            Event LS_COMPUTE_INTERFACE_NORMAL_VELOCITY

         End
       End
       Begin Parameters For Transient Stepper
            Begin Parameters For Aria Region Fluid_Region
             Initial Time Step Size = {dt = 0.1}
               #Minimum Resolved Time Step Size = {dt}
               Minimum Time Step Size = {dt/1000}
             #Time Step Variation    = Adaptive
               #PREDICTOR-CORRECTOR TOLERANCE = 0.001
             #Courant Limit = 0.2
             Time Step Variation    = Fixed
         End
           Begin Parameters For Aria Region LS_Region
            #Initial Time Step Size = {1000*dt}
            Initial Time Step Size = {dt}
            Time Step Variation    = Fixed
         End
       End
      End

      Begin Transfer Fluid_to_LS
         Copy Volume Nodes From Fluid_Region To LS_Region
         Send Field solution->VELOCITY State New To solution->VELOCITY State
New
         Send Field solution->VELOCITY State old To solution->VELOCITY State
old
      End
      Begin Transfer LS_To_Fluid
         Copy Volume Nodes From LS_Region To Fluid_Region
         Send Field solution->LEVEL_SET State New To solution->LEVEL_SET
State New
         Send Field solution->LEVEL_SET State old To solution->LEVEL_SET
State old
      End
      Begin Transfer PROJ_LS_To_Fluid
         Copy Volume Nodes From LS_Region To Fluid_Region
         Send Field solution->PROJECTED_Level_Set_Curvature State New To
solution->PROJECTED_Level_Set_Curvature State New
         Send Field solution->PROJECTED_Level_Set_Curvature State old To
solution->PROJECTED_Level_Set_Curvature State old
      End

      Begin Aria Region LS_Region

         Use Linear Solver iterative
         use Finite Element Model foam

         Nonlinear Solution Strategy    = Newton
```

```
         Maximum Nonlinear Iterations   = 5
         Nonlinear Residual Tolerance   = 1e-6
         #Nonlinear Correction Tolerance = 1e+12 # LINEAR
         Nonlinear Correction Tolerance = 1e-6
         Nonlinear Relaxation Factor    = 1.0

         EQ Level_Set for Level_Set on block_1 Using Q1 with Mass Adv
Taylor_Galerkin
         #IC Linear on block_1 Level_Set COEFF = -11.9 0.  1.0  0.
         IC Linear on block_1 Level_Set COEFF = -1.13 0.  1.0  0.


         BC Disting for Level_Set on surface_1 = Polynomial
Variable=Dt_Level_Set Order=1 C1=1

         EQ Momentum for Velocity on block_1 using Q1 with XFER
         Predictor Fields = Not Velocity

       EQ Lumped_Div_Projection for Projected_Level_Set_Curvature on block_1
Using Q1 with Def
         Predictor Fields = Not Projected_Level_Set_Curvature

         BC Flux for Lumped_Div_Projection on surface_1 = Curvature
       BC Flux for Lumped_Div_Projection on surface_3 = Curvature

         BEGIN LEVEL SET INTERFACE LS
           Distance Variable = solution->LEVEL_SET
           Velocity Variable = solution->VELOCITY
           NARROW BAND WIDTH = {3.0*alpha}
         END LEVEL SET INTERFACE LS

     end aria region LS_Region

      begin aria region Fluid_Region

       #use linear solver iterative
         #use Linear Solver SUPERLU
       use linear solver gmres_ilut
       #use linear solver gmres_ilu
       use Finite Element Model foam

         nonlinear solution strategy    = Newton
         nonlinear residual tolerance     = 1e-8
         nonlinear correction tolerance   = 1e-6
         nonlinear relaxation factor      = 1.0
       Maximum Nonlinear Iterations   = 10

         EQ Level_Set for Level_Set on block_1 Using Q1 with XFER
         EQ Lumped_Div_Projection for Projected_Level_Set_Curvature on
block_1 Using Q1 with XFER
       Predictor Fields = Not Level_Set
         Predictor Fields = Not Projected_Level_Set_Curvature

       EQ continuity_A for pressure on block_1 using Q1 with mass adv
         EQ continuity_B for pressure on block_1 using Q1 with div

       Predictor Fields = Not Pressure
```

```
        PRESSURE STABILIZATION is PSPP_TRANSIENT with scaling = 0.15

        EQ momentum_A for Velocity on block_1 using Q1 with Diff Mass Adv
Src SUPG
        EQ momentum_B for Velocity on block_1 using Q1 with Diff Mass Adv
Src SUPG

      # width = {width = 1.0*alpha}

        mesh group slip_boundary = surface_1 surface_3
        #BC Flux for Momentum_A on slip_boundary = Oriented_Slip
Beta_Normal=1.e-7 Beta_Tangent=1.e-1
        #BC Flux for Momentum_B on slip_boundary = Oriented_Slip
Beta_Normal=1.e-7 Beta_Tangent=1.e-1


        BC Flux for Momentum_A on slip_boundary = Slip Beta= {beta}
        BC Flux for Momentum_B on slip_boundary = Slip Beta= {beta}
        BC Rotated for Momentum_A on slip_boundary = Kinematic
        BC Rotated for Momentum_B on slip_boundary = Kinematic


      Source For Momentum_A on block_1 = Hydrostatic gx = 0 gy = {-g} gz = 0
      Source For Momentum_B on block_1 = Hydrostatic gx = 0 gy = {-g} gz = 0

        SOURCE FOR momentum_A ON block_1 = LS_CAPILLARY
        SOURCE FOR momentum_B ON block_1 = LS_CAPILLARY

        # ----------- ENERGY EQUATION FORMULATION -------------------------
----

        EQ Energy_A For Temperature On block_1 Using Q1 With Mass Adv Diff
Src
        #SOURCE for Energy_A on block_1 = CURING_FOAM_HEAT_OF_RXN
RHO_INITIAL = {rhof_i} RHO_FINAL={rhof_f} RHO_LIQUID = {rho_liq}  RHO_GAS =
{rho_gas} RHO_EPOXY = {rho_liq} H_rxn = {hrxn}
      SOURCE for Energy_A on block_1 = CURING_FOAM_HEAT_OF_RXN H_rxn =
{hrxn}   Mass_Fraction_epoxy={m_epoxy}  extent_subindex=0
      #SOURCE for Energy_A on block_1 = Curing_Foam_Latent_Heat
        SOURCE for Energy_A on block_1 = CONSTANT value = 0.0

        EQ Energy_B For Temperature On block_1 Using Q1 With Mass Adv Diff
Src
        SOURCE for Energy_B on block_1 = CONSTANT value = 0.0

        IC Const on block_1 Temperature = {T_i}

        #BC Flux for Energy_A on surface_1 = Nat_Conv T_ref={T_oven} H={h}
        #BC Flux for Energy_B on surface_1 = Nat_Conv T_ref={T_oven} H={h}
        #BC Flux for Energy_A on surface_2 = Nat_Conv T_ref={T_oven} H={h}
        #BC Flux for Energy_B on surface_2 = Nat_Conv T_ref={T_oven} H={h}
        #BC Flux for Energy_A on surface_3 = Nat_Conv T_ref={T_oven} H={h}
        #BC Flux for Energy_B on surface_3 = Nat_Conv T_ref={T_oven} H={h}


        BC Const Dirichlet at surface_3 Temperature =  {T_oven}
```

```
         BC Const Dirichlet at surface_2 Temperature =  {T_oven}
         BC Const Dirichlet at surface_1 Temperature =  {T_oven}

         # ----------- EXTENT OF RXN EQUATION FORMULATION ------------------
---------
         #
         EQ Species_0_A for Species_0 on block_1 using Q1 with Mass Adv diff
Src Taylor_Galerkin
         SOURCE for Species_0_A

         Postprocess Volume_Fraction_Gas on block_1
         Postprocess Bulk_Viscosity on block_1

       Begin Results Output fluid output
            database Name = soln.e
            At Step 1, Increment is 20
            Title Aria 3D Mold Filling
          Nodal Variables = solution->Velocity  as V
          Nodal Variables = residual->Velocity  as RV
          Nodal Variables = solution->Level_Set as F
            Nodal Variables = solution->Pressure as P
            Nodal Variables = residual->Pressure as RP
          Nodal Variables = solution->Projected_Level_Set_Curvature  as GradF
            Nodal Variables = solution->Temperature as T
            Nodal Variables = residual->Temperature as RT
            Nodal Variables = solution->Species_0 as x0
            Nodal Variables = residual->Species_0 as Rx0
            Nodal Variables = pp->Div_Velocity as DV
          Nodal Variables = pp->Density as rho
            Nodal Variables = pp->Viscosity as MU
            Nodal Variables = pp->thermal_conductivity as k
            Nodal Variables = pp->specific_heat as Cp
          Nodal Variables = pp->Volume_Fraction_Gas as vol
          Nodal Variables = pp->Bulk_Viscosity as kappa
       End

      end aria region Fluid_Region

   end procedure myProcedure

end sierra myJob
```

DISTRIBUTION

4 James Mahoney
Advanced Engineering Simulation and Analysis
Honeywell Federal Manufacturing and Technologies, Kansas City Plant
P.O. Box 419159
Kansas City, Missouri 64141-6159

1	MS0346	Christopher Brotherton	1513
1	MS0346	Anne Grillet	1513
1	MS0346	Lisa Mondy	1514
1	MS0346	John Torczynski	1513
1	MS0346	Dan Rader	1513
1	MS0382	David Noble	1514
1	MS0382	Patrick Notz	1541
1	MS0382	Basil Hassan	1541
1	MS0557	Sarah Leming	1521
1	MS0555	Chris O'Gorman	1522
1	MS0824	Michael Prairie	1510
1	MS0826	Mike Valley	1512
1	MS0828	Amalia Black	1544
1	MS0836	Terry Aselage	1514
1	MS0836	Andrew Kraynik	1514
1	MS0836	Harry Moffat	1516
2	MS0836	Rekha Rao	1514
1	MS0886	James Aubert	1821
1	MS0888	Edward Russick	1821
1	MS0958	Mike Kelly	2453
1	MS1064	Marc Polosky	2614
1	MS1245	Douglas Adolf	2453
1	MS1411	Mat Celina	1821
1	MS1411	Jamie Kropka	1821
1	MS0899	Technical Library	9536 (electronic copy)


Sandia National Laboratories